DOCUMENT RESUME

ED 076 419 SE 016 063

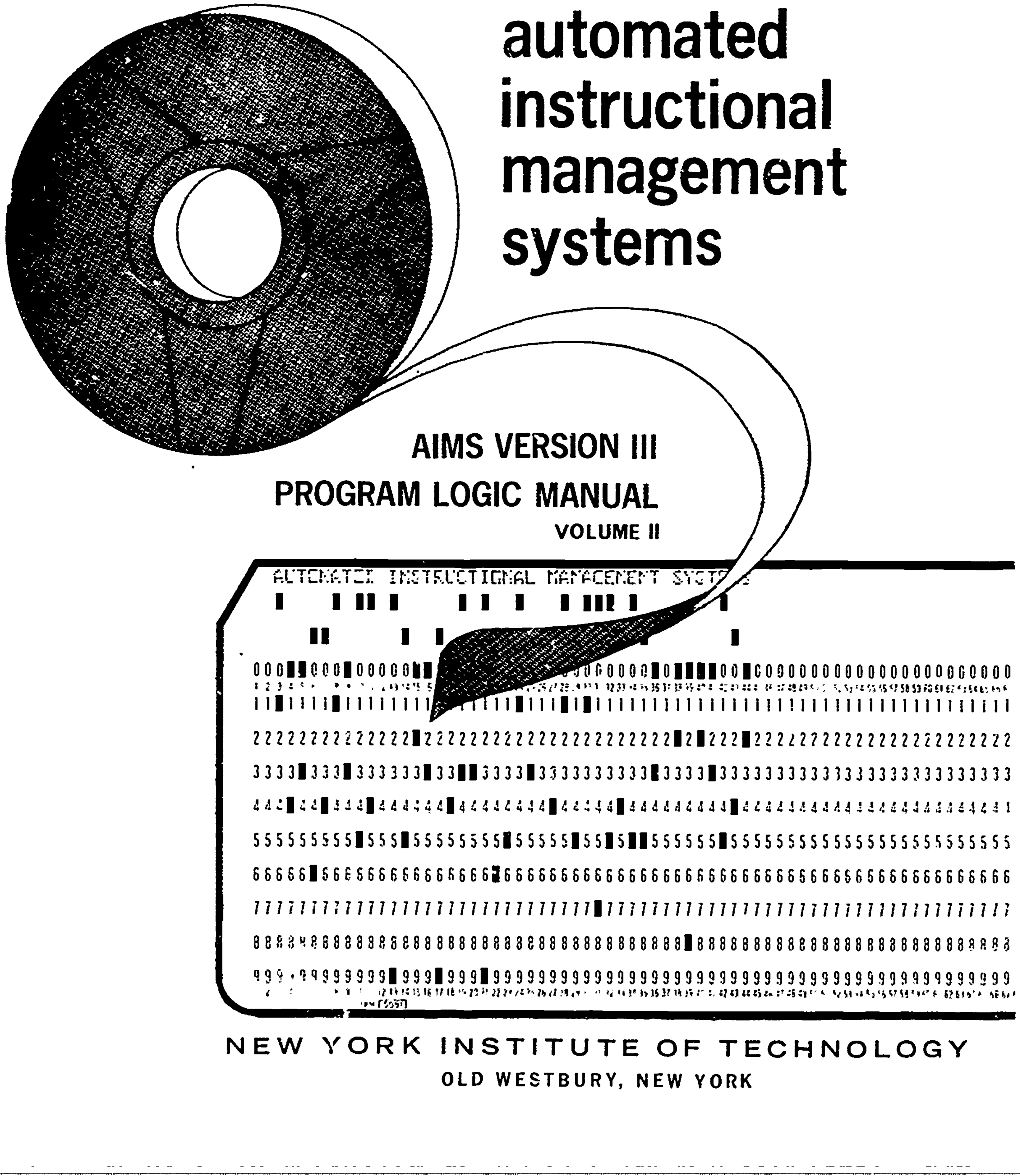
TITLE	Automated Instructional Management Systems (AIMS) Version III, Program Logic Manual, Volume 2. Source Statement Listings.
INSTITUTION	New York Inst. of Tech., Old Westbury.
SPONS AGENCY	Office of Education (DHEW), Washington, D.C. Bureau of Research.
BUREAU NO	BR-8-0157
PUB DATE	[73]
CONTRACT	OEC-0-8-080157-3691(010)
NOTE	106p.
EDRS PRICE	MF-$0.65 HC-$6.58
DESCRIPTORS	*Computer Assisted Instruction; Computer Programs; *Computer Science; Instruction; *Instructional Media; Instructional Technology; Management Information Systems; Mathematics Education; *Programed Materials
IDENTIFIERS	*Automated Instructional Management System

ABSTRACT

This second program logic manual for the Automated Instructional Management System - Version III contains source statement listings for 23 programs of the system. For related materials, see SE 016 059 through SE 016 064. (DT)

ADVANCED
SYSTEMS
LABORATORY

automated instructional management systems



AIMS VERSION III
PROGRAM LOGIC MANUAL
VOLUME II

NEW YORK INSTITUTE OF TECHNOLOGY
OLD WESTBURY, NEW YORK

AUTOMATED INSTRUCTIONAL MANAGEMENT SYSTEM

PROGRAM LOGIC MANUAL

SOURCE STATEMENT LISTINGS

Prepared by the Staff of
The Advanced Systems Laboratory

Ernest N. O'Dierno, Director

FOREWORD

- The Automated Instructional Management System (AIMS) was designed to monitor, score, and evaluate individual students, groups of students, and curricular content in a course environment designated for educational management.

- The AIMS System was designed around IBM System/360, and Version III was generated with Model 30/Release 20 IBM Disk Operating System (DOS).

- All source statement listings contained in this manual have been developed with U. S. Office of Eduction funds under Research Contract No. OEC-0-8-080157-3691(010).

ADVANCED SYSTEMS LABORATORY
New York Institute of Technology
Old Westbury, L.I., New York

ADVANCED SYSTEMS LABORATORY A370-670
AIMS III PROGRAM LOGIC MANUAL Volume 2
SOURCE STATEMENT LISTINGS

TABLE OF CONTENTS

Page

CTIOMAIN

UTILITY ROUTINES

AHEWMAIN

UTILITY ROUTINES

AIMSHEAD,*

AIMSCLAS, AIMSHEAD

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

```
        I S III 360 DOS STIMAN LIST1

// JOB 39413         CTIOMAIN
// ASSGN SYSLNK,X'192'
// DLBL IJSYSLN,'SYSLNK',66/365,SD
// EXTENT SYSLNK,000002,1,0,10,1000
// OPTION CATAL
   PHASE CTIOMAIN,*
// EXEC FFORTRAN
      DIMENSION ITEST(3)
      DIMENSION ICOM(46)
      INTEGER HEADER,SYSTEM
      INTEGER ICNTRL(15)
      INTEGER*2 LPERM,NPERM,NOS(50),PIRS(50),NEXREC(65)
      DEFINE FILE 5(1500,33,U,I5),
                  6( 800,33,U,I6),
     3            7( 200,100,U,I7),
     4           8(4000,35,U,I8),
     5            9( 250,35,U,I9),
     6          10(8040,33,U,I10),
     7           11( 400,23,U,I11),
     8           12( 100,25,U,I12)
      INTEGER*2 IVECT(4,10,40)
      COMMON/CHKREC/IVECT
      COMMON /SYSTEM/ NLESS,NDECK,NREX,NQUEST,NSTUD
      COMMON /FILES/ IFILES(15)
      EQUIVALENCE (IFILES(1),ICD),(IFILES(3),IPT), (IFILES(6),HEADER),
     2(IFILES(12),SYSTEM)
      INTEGER M1/'((CA'/,M2/'RK S'/,M3/'FNSH'/
      DATA IT/'((CA'/,IS/'RDS '/,IHE/'HEAD'/,ISTU/'STUD'/,ITP/'NOTA'/
      LOGICAL*1 FIRST
      DO 5 IDEV=1,15
    5 IFILES(IDEV)=IDEV
C
      NLESS=40
      NSTUD= 185
      NREX=4
      NQUEST=48
      NDECK=10
 1000 DO 1 N=1,15
    1 ICNTRL(N)=0
      ICNTRL(6)=0
      ICNTRL(7)=10
      ICNTRL(8)=14
      ICD=1
      IPT=3
      READ (ICD,2) (ITEST(N),N=1,3),ICNTRL(1),ICNTRL(3),ICNTRL(5),ICON,
     2ICOM
    2 FORMAT (3A4,5X,I2,1X,I1,1X,I2,1X,A4,7X,46A1)
      IF ((ITEST(1).EQ.IT).AND.(ITEST(2).EQ.IS)) GO TO 100
      IF(ITEST(1).EQ.M1.AND.ITEST(2).EQ.M2.AND.ITEST(3).EQ.M3) GO TO 110
    4 WRITE (IPT,3)
      WRITE(IPT,2) (ITEST(N),N=1,3),ICNTRL(1),ICNTRL(3),ICNTRL(5),ICON,
     2ICOM
    3 FORMAT (1H0,/1H0,'***** ILLEGAL CONTROL CARD *****',//////////////)
      GO TO 1000
  100 IF (ITEST(3).EQ.ISTU) ICNTRL(4)=2
      IF (ITEST(3).EQ.IHE) ICNTRL(4)=1
      IF (ICNTRL(4).EQ.0) GO TO 4
      IF (ICNTRL(1).LE.0) GO TO 4
      IF ((ICNTRL(3).GT.4).AND.(ICNTRL(5).NE.5)) GO TO 4
      ICNTRL(2)=2
      IF (ICON.EQ.ITP)  ICNTRL(2)=1
      GO TO 3000
C MARK SENSE CONTROL SECTION
  110 IF(ICNTRL(1).LT.0) GO TO 4
C PRE&POST TEST FORM
      IF(ICNTRL(3).EQ.1) ICNTRL(4)=3
C STUDY GUIDE FORM
      IF(ICNTRL(3).EQ.2) ICNTRL(4)=4
C BOTH FORMS
      IF(ICNTRL(3).EQ.3) ICNTRL(4)=5
      IF(ICNTRL(4).EQ.0) GO TO 4
      ICNTRL(2)=2
      IF(ICON.EQ.ITP) ICNTRL(2)=1
 3000 IF (ICNTRL(4).EQ.1) GO TO 4000
      READ (SYSTEM'2) LPERM,NPERM
      READ (SYSTEM'3) PTRS
      READ (SYSTEM'4) NOS
      DO 3003 ICLR=1,40
      DO 3003 ICLS=1,10
      DO 3003 ICLT=1,4
 3003 IVECT(ICLT,ICLS,ICLR)=-1
      DO 3001 ILESS=1,LPERM
      ISTART=PTRS(ILESS)
      IEND=ISTART+NOS(ILESS)-1
      IDIFF=ISTART-1
      DO 3002 ISPOT=ISTART,IEND
      READ (HEADER'ISPOT)HEDREC
      IVECT(1,ISPOT-IDIFF,ILESS)=HEDREC(3)
      IVECT(2,ISPOT-IDIFF,ILESS)=HEDREC(4)
      IVECT(3,ISPOT-IDIFF,ILESS)=HEDREC(10)
      IVECT(4,ISPOT-IDIFF,ILESS)=HEDREC(6)
 3002 CONTINUE
 3001 CONTINUE
C                SO MUCH FOR THAT
 4000 FIRST=.TRUE.
 4001 FORMAT (1H1,/1H0,3A4,3X,I2,1X,I1,1X,I2,1X,A4,7X,46A1,//)
      WRITE (IPT,4001) ITEST,ICNTRL(1),ICNTRL(3),ICNTRL(5),ICON,IC
      IDUM = ICNTRL(4)
      GO TO(210,210,220,230,220),IDUM
      GO TO 4
  210 CONTINUE
      CALL CARDS(ICNTRL,FIRST)
      GO TO 250
  220 CONTINUE
      CALL MARKPT(ICNTRL(1),ICNTRL(2))
      IF(ICNTRL(4).EQ.5) GO TO 230
      GO TO 250
  230 CALL MARKSG(ICNTRL(1),ICNTRL(2))
  250 CONTINUE
      CALL EXIT
      END
      SUBROUTINE MARKPT(ICRSE,NOTAPE)
      INTEGER*2 IVECT(4,10,40),JCRSE,JSTUD,CNTREC
       INTEGER*2 ICSN(3),IMINS(2),ISEG(2),IHRS,IVOL(2), ICSE(4),ITYPE
       INTEGER*2 IANS(48),QMRK,QTST/'Q'/,ID(3)/3*'  '/,SEG/1/
       INTEGER*2 OUTPUT(65),CSN,MINS,SEG,VOL,CSE,TYPE,ANS(48),NUMGOD
       INTEGER*2 NUMQST,TPTEST(5)/'A','C','E','G','I'/,JTP,NUMBAD
       INTEGER*2 CODE(6)/0,2,4,8,16,32/
 AIMS III SOURCE STATEMENT LISTING

      INTEGER*2 TYPCODE(5)/3,5,0,2,1/,NVCL
      COMMON/CHKREC/IVLCT
      COMMON/SYSTEM/NLESS,NBLCK,NSEX,NQUEST,NSTUD
      COMMON/FILES/IFILES(15)
     0EQUIVALENCE (IFILES(13),MRKINP),(IFILES(3),IOUT),(IFILES(14),INT),
     1(OUTPUT(1),VOL) , (OUTPUT(2),CSN) , (OUTPUT(3),SEG),
     2(OUTPUT(4),TYPE) , (OUTPUT(5),CSE) , (OUTPUT(7),ID(1)),
     3(OUTPUT(12),NUMQST) , (OUTPUT(16),IHRS) , (OUTPUT(17),MINS) ,
     4(OUTPUT(18),ANS(1)) , (OUTPUT(6),SEQ)
     0INTEGER*2 MAPANS(48)/1,6,11,16,21,2,7,12,17,22,3,8,13,18,23,4,9,
     114,19,24,5,10,15,20,25,26,31,36,41,46,27,32,37,42,47,28,33,38,43,
     248,29,34,39,44,30,35,40,45/
      CNTREC=0
      JCRSE=ICRSE*1
      JSTUD=NSTUD*1
      NVOL=NLESS*1
      NUMBAD=0
      NUMQST=0
      WRITE(IOUT,610)
  6100FORMAT(1H1,10X,'AI*S MARK SENSE PROCESSING FOR PRE OR POST TEST FO
     1RM (DC6007).')
 1020 CONTINUE
      DO 1030 J=1,5
 1030 OUTPUT(J)=0
      DO 1040 J=10,65
 1040 OUTPUT(J)=0
     0READ(MRKINP,10,END=9999) ICSN(1),IMINS(1),ISEG(1),ICSN(2),IMINS(2)
     1,ISEG(2),ICSN(3),IHRS,IVOL(1),IVOL(2),ICSE(1),ITYPE,ICSE(2),
     2ICSE(3),ICSE(4),(IANS(MAPANS(J)),J=1,48),QMRK
   10 FORMAT(1111,A1,5111,A1)
      CNTREC=CNTREC+1
      IF(QMRK.EQ.QTST) GO TO 1000
      NUMBAD=NUMBAD+1
      WRITE(IOUT,500) CNTREC
  5000FORMAT('  *** ERROR *** PRE OR POST TEST FORM (DC6007) HAS BAD QUA
     1LITY. RECORD IGNORED.   RECORD NO., ',I4,/,16X,'CHECK FOR ERASORS,
     2SMEARS,AND DOUBLE MARKS',//)
      GO TO 1020
 1000 CONTINUE
C CONVERT TEST ID DATA
      CSN = ICSN(3)*100 + ICSN(2)*10 + ICSN(1)
      MINS = IMINS(2)*10 + IMINS(1)
      SEG  = ISEG(2)*10 + ISEG(1)
      VOL  = IVOL(2)*10 + IVOL(1)
      CSE  = ICSE(4)*1000 + ICSE(3)*100 + ICSE(2)*10 + ICSE(1)
      DO 1050 JTP=1,5
      IF(ITYPE.EQ.TPTEST(JTP)) TYPE=TYPCODE(JTP)
 1050 CONTINUE
C TEST  ID DATA FOR VALIDITY
      IF(CSN.GT.0.AND.CSN.LE.JSTUD) GO TO 2000
      NUMBAD=NUMBAD+1
      CALL ERRHED(OUTPUT,ICRSE,CNTREC)
      WRITE(IOUT,510)
  510 FORMAT(' COURSE STUDENT NUMBER IN ERROR. ABOVE WORK REJECTED.',/)
      GO TO 1020
 2000 CONTINUE
      IF(VOL.GT.0.AND.VOL.LE.NVOL) GO TO 2020
      NUMBAD=NUMBAD+1
      CALL ERRHED(OUTPUT,ICRSE,CNTREC)
      WRITE(IOUT,530) NLESS
  530FORMAT(' VOLUME NUMBER TOO LARGE, LIMIT SET AT, ',I3,'. ABOVE WORK
     1REJECTED,',/)
      GO TO 1020
 2020 CONTINUE
      DO 1060 JDECK=1,10
      IF(IVECT(1,JDECK,VOL).NE.SEG) GO TO 1060
      IF(IVECT(2,JDECK,VOL).NE.TYPE) GO TO 1060
      GO TO 2010
 1060 CONTINUE
      NUMBAD=NUMBAD+1
      CALL ERRHED(OUTPUT,ICRSE,CNTREC)
      WRITE(IOUT,520)
  520FORMAT(' VOLUME,SEGMENT,OR TYPE OF WORK NOT PRESENT IN AIMS FILES'
     1/,' PLEASE CHECK FILES. ABOVE WORK REJECTED.',/)
      GO TO 1020
C GOOD RECORDS MAKE IT TO HERE
 2010 DO 3000 JANS=1,48
      IF(IANS(JANS).NE.0) NUMQST=NUMQST+1
      ANS(JANS) = CODE(IANS(JANS)+1)+1
 3000 CONTINUE
      IF(NOTAPE.EQ.1) GO TO 3010
      WRITE(IRT) OUTPUT
 3010 GO TO 1020
 9999 CONTINUE
      NUMGOD= CNTREC-NUMBAD
      WRITE(IOUT,600) NUMGOD,NUMBAD
  600FORMAT(1H0,10X,' AIMS MARK SENSE PROCESSING FOR PRE OR POST TEST(J
     1C0007) COMPLETE.',/,20X'NUMBER OF RECORDS ACCEPTED, ',I4,
     2                    /,20X'NUMBER OF RECORDS REJECTED, ',I4  )
      RETURN
      END
      SUBROUTINE ERRHED(RECD  ,ICRES,RECCNT)
      COMMON/FILES/ IFLE(2),IPT
      INTEGER*2 RECD(17)  ,ICRES,SEQ,JTYPE,INFORM(12),RECCNT
      JTYPE=2
      SEQ = 0
  102 CALL INFO(INFORM)
      WRITE (IPT,103) INFORM,RECCNT  ,SEQ,RECD(6),JTYPE
  103 FORMAT (1H0,/1H ,'**** CARD IN ERROR **** JOB NAME ',4A2,5X,'DATE-
     2 ',4A2,5X,'TIME- ',4A2,'  *********PHYSICAL RECORD - ',I8,/1H ,
     3'PREVIOUS SEQUENCE NO.= ',I2,' CARD SEQUENCE NO.= ',I2,'  TYPE OF
     4 CARDS = ',I1,' (1=HEADER,2=STUDENT    ) ' )
      WRITE (IPT,117) ICRES   ,RECD(5),RECD(1),RECD(3),RECD(4)
  117 FORMAT (1H ,'COURSE BEING PROCESSED = ',I2,' COURSE NUMBER ON CARD
     2 = ',I2,' LESSON = ',I2,' SEGMENT = ',I2,' TYPE = ',I2)
      IF (JTYPE.EQ.1) WRITE (IPT,118) RECD(13),RECD(11),(RECD(N),N=7,9),
     2RECD(10)
  118 FORMAT (1H ,'NUMBER OF QUESTIONS = ',I2,' NUMBER OF SELECTIONS = '
     2,I3,' I.D.FIELD = ',3A2,' (S) OR (D) = 'A1)
      IF (JTYPE.EQ.2) WRITE (IPT,119) RECD(2),(RECD(N),N=7,9),RECD(14),
     2RECD(15),RECD(13),RECD(16),RECD(17)
  119 FORMAT (1H ,'COURSE STUDENT NUMBER = ',I3,' STUDENT I.D. NUMBER =
     2 '3A2,' DATE ON CARD = ',I2,'/',I2,'/',I2,' TIME ON CARD = ',I2,
     3'.',I2)
      WRITE (IPT,121)
  121 FORMAT (1H ,'------------------ E R R O R S ------------------
     2-------------------------------------------------------',/)
      RETURN
      END
      SUBROUTINE MARKSG(ICRSE,NGTAPE)
C     THIS PROGRAM WILL READ SCANNER TAPE AND UNSCRAMBLE FORM 0C6006
      INTEGER*2 IN(121),OUT(65),BLANK,TYPE(6),TYPE1(6),MARK(5),FF(35),
     1ERSP,CONF,ERR
      INTEGER*2 IVECT(4,10,40),BLANK2/'  '/
      INTEGER*2 NUMGOD,NUMBAD,JSTUD,JLESS
      COMMON/SYSTEM/NLESS,NDECK,NREX,NQUEST,NSTUD
      COMMON/CHKREC/IVECT
      DATA BLANK/' '/,TYPE/'A','C','E','G','I','K'/,MARK/3,5,9,17,33/,
     1 TYPE1/5,5,7,2,1,6/
      JSTUD=NSTUD*1
      JLESS=NLESS*1
      NUMGOD=0
      NUMBAD=0
      I3 = 4
      I4 = 14
      OUT(5) = 1
      OUT(13)=0
      OUT(14)=0
      OUT(15)=0
      OUT(7) = BLANK2
      OUT(8) = BLANK2
      OUT(9) = BLANK2
      OUT(10)=0
      OUT(11)=0
      IREC = 0
      WRITE(3,600)
  6000FORMAT(1H1,10X,'AIPS MARK SENSE PROCESSING FOR STUDY GUIDE FORM
     1(0C6006).')
C
C---- READ SCANER TAPE
C
   23 READ(13,1,END=20)(IN(I),I=1,24),(IN(J),IN(J+1),IN(J+2),IN(J+3),
     X IN(J+48),IN(J+49),IN(J+50),IN(J+51),J=25,69,4),IN(121)
    1 FORMAT(32A1,8X,8A1,8X,81A1)
      IREC = IREC + 1
      IERR = 0
      IF(IN(121)+ 10176) 70,71,70
   70 WRITE (3,10) IREC
   10 FORMAT (' RECORD ',I3,' HAS BAD QUALITY--RECORD IGNORED '/)
      GO TO 23
C
C---- CHECK FOR BLANK STUDENT COURSE NUMBER FIELD
C
   71      IF(LOOK(IN(01))+ LOOK(IN(04))+ LOOK(IN(07))) 21,24,21
   21 WRITE (3,2) IREC
    2 FORMAT (' RECORD ',I3,' HAS A BLANK IN STUDENT COURSE NUMBER FIELD
     1' )
      IERR = IERR + 1
      GO TO 123
   24 OUT (2) =  (4032+IN(07) ) / 256 *100
     1           +(4032+IN(04) ) / 256 *10
     2           +(4032+IN(01) ) / 256
      IF(OUT(2).LE.JSTUD) GO TO 123
      IERR=IERR+1
      WRITE(3,540) IREC
  5400FORMAT(' RECORD ',I3,' COURSE STUDENT NUMBER TOO LARGE--RECORD IG
     1NORED '/)
```

```
C
C---- CHECK FOR BLANKS IN COMPLETION TIME FIELD
C
  123 IF(LOOK(IN(2)) + LOOK(IN(5)) + LOOK(IN(8)) ) 26,28,26
   26 WRITE (3,3) IREC
    3 FORMAT (' RECORD ',I3,' HAS BLANKS IN COMPLETION TIME FIELD--FIELD
     1SET TO ZERO' )
      OUT(16) = 0
      OUT(17) = 0
      GO TO 29
   28 OUT(16) = (4032+IN(8)) / 256
      OUT(17) = (4032+IN(5)) / 256 * 10
     1          +(4032+IN(2)) / 256
C
C---- CHECK FOR BLANKS IN SEGMENT FIELD
C
   29 IF(LOOK(IN(14))                    ) 30,31,30
   30 WRITE (3,4) IREC
    4 FORMAT (' RECORD ',I3,' HAS BLANKS IN SEGMENT FIELD ' )
      IERR = IERR + 1
      OUT(3) = 0
      GO TO 32
   31 OUT(3) =(4032+IN(14))/256
C
C---- CHECK FOR BLANKS IN VOLUME FIELD
C
   32 IF(LOOK(IN(18))+LOOK(IN(21)) ) 33,34,33
   33 WRITE (3,5) IREC
    5 FORMAT (' RECORD ',I3,' HAS BLANKS IN VOLUME FIELD' )
      IERR = IERR + 1
      GO TO 35
   34 OUT(1)  =  (4032+IN(21))/ 256 * 10
     1           +(4032+IN(18))/ 256
      IF(OUT(1).LE.JLESS) GO TO 35
      IERR=IERR+1
      WRITE(3,550) IREC
  550 FORMAT(' RECORD ',I3,' VOLUME TOO LARGE-- RECORD IGNORED '/)
C
C---- TEST TYPE FIELD
C
   35 OUT(4) = 4
C
C---- TEST COURSE NUMBER FIELD
C
      IF(LOOK(IN(3))+LOOK(IN(6))+LOOK(IN(9))+LOOK(IN( 12)))40,41,40
   40 WRITE (3,16) IREC
   16 FORMAT ( ' RECORD ',I3,' HAS COURSE NUMBER TROUBLE ')
      OUT(5) = 21
      GO TO 42
   41 OUT(5) = (4032+IN(12)) / 256 *1000
     1         +(4032+IN(09)) / 256 *100
     1         +(4032+IN(06)) / 256 *10
     3         +(4032+IN(03)) / 256
      DO 1060 JDECK=1,10
      IF( IVECT(1,JDECK,OUT(1)).NE.OUT(3)) GO TO 1060
      IF( IVECT(2,JDECK,OUT(1)).NE.OUT(4)) GO TO 1060
      GO TO 2010
 1060 CONTINUE
      WRITE(3,520) IREC
  520 FORMAT(' RECORD, ',I5,'VOLUME,SEGMENT,OR TYPE OF WORK NOT PRESENT
     1IN AIMS FILES.'/,10X,'PLEASE CHECK FILES. ABOVE WORK REJECTED.')
      GO TO 23
 2010 CONTINUE
C
C---- COUNT NUMBER OF QUESTIONS ANSWERED
C
   42 KOUNT = 0
      OUT(12) = KOUNT
      DO 60 I=25,117,4
      K = I+3
      DO 61 J = I,K
      IF (IN(J)-16448) 60,61,60
   61 CONTINUE
      KOUNT = KOUNT + 1
   60 CONTINUE
      OUT(12) = 24 - KOUNT
C
C             ZERO OUTPUT AND SET RESPONSE
C
      DO 67 I =16,65
   67 OUT(I)=0
      JJ = 17
      DO 62 I =25,117,4
      K = I + 3
      L = 0
      JJ = JJ + 1
      DO 63 M = I,K
      L = L + 1
      IF(IN(M)-16448)64,63,64
   64 OUT(JJ) = OUT(JJ) + 2**L
   63 CONTINUE
      OUT(JJ) = OUT(JJ) + 1
   62 CONTINUE
C
C---- DONE -- WRITE OUT RESPONSE TAPE AND GO TO READ NEW RECORD
C
      IF(IERR) 55,56,55
   55 WRITE (3,8) IREC,IERR
    8 FORMAT (' RECORD ',I3,' HAS ',I3,' UNRECOVERABLE ERRORS--RECORD IG
     1NORED ' //)
      GO TO 23
   56 CONTINUE
      NUMGOD=NUMGOD+1
      IF(NOTAPE.EQ.1) GO TO 23
      WRITE (I4) OUT
      GO TO 23
   20 WRITE(3,9)
    9 FORMAT(1H0,10X,'AIMS MARK SENSE PROCESSING FOR STUDY GUIDE FORM
     1(DC3006) IS COMPLETE.')
      NUMBAD=IREC-NUMGOD
      WRITE(3,530) NUMGOD,NUMBAD
 5300 FORMAT(20X,'NUMBER OF RECORDS ACCEPTED, ',I4,
     1     /,20X,'NUMBER OF RECORDS REJECTED, 'I4)
      RETURN
      END
      FUNCTION LOOK(ITEST)
      INTEGER*2 ITEST,DATA(10)
      DATA DATA/'0','1','2','3','4','5','6','7','8','9'/
```

FILMED FROM BEST AVAILABLE COPY

```
        AIDS III SOURCE STATEMENT LISTING

          LOOK  = 0
          DO 20 I=1,10
          IF(ITEST-DATA(I)) 20,21,20
       20 CONTINUE
          LOOK = 1
       21 RETURN
          END
          SUBROUTINE CARDS(ICNTRL,FIRST)
          LOGICAL*1 FIRST
          LOGICAL*1 CHARLY
          DIMENSION ICNTRL(15)
          LOGICAL*1 ACTION
          INTEGER RIGHT,LEFT
          INTEGER*2 IVECT(4,10,40)
          INTEGER*2 AREA(80),RECD(45)
          INTEGER*2 QCOLS(12),NUMCOL(23)
          INTEGER*2 SHIFTS
          INTEGER*2 ID1,ID2,ID3,ID4,ID5
          DATA QCOLS/4,10,16,22,28,34,40,46,52,58,64,70/
          DATA NUMCOL/2,6,8,12,14,18,20,24,26,30,32,36,38,42,44,48,50,54,56,
         260,62,66,68/
          COMMON /FILES/ ICD,IPC,IPT,IWI,ISCH,HEADER,IDCT,IQUST,IBKUD,ISCORE
         2,ITEXT,SYSTEM
          COMMON/CHRREC/IVECT
          IPT=3
          ITAPE=ICNTRL(8)
          ITAPE=14
          LEFT=1
          RIGHT=2
          ICODE=ICNTRL(4)
          IF (FIRST)          ASSIGN 1 TO ILOG
          IF (.NOT.FIRST)              ASSIGN 2000 TO ILOG
          GO TO ILOG,(1,2000)
        1 REWIND ITAPE
          NREX=0
        2 CALL CRLIN(AREA)
          CHARLY=.FALSE.
          ACTION=.FALSE.
          IF ((AREA(1).EQ.768).AND.(AREA(2).EQ.1058))GO TO 999
          IF ((AREA(1).EQ.768).AND.(AREA(2).EQ.768)) GO TO 1000
          IF ((AREA(1).EQ.768).AND.(AREA(2).EQ.2048)) GO TO 1000
          IF ((AREA(1).EQ.2066).AND.(AREA(2).EQ.2066)) GO TO 1000
          GO TO 99999
 1000     DO 1002 N=1,80
          AREA(N)=LETTER(AREA(N))
 1002     CONTINUE
 1001     FORMAT (1H1,/1H1,'***** JOB STACK ERROR ******************',/)
          WRITE (IPT,1001)
          WRITE (IPT,1003) AREA
 1003     FORMAT (1H ,'THIS CARD READ IN BINARY- ',80A1,/1H ,
         2 ' REPAIR JOBSTACK AND CANCEL THIS JOB ********************',/////
         3///)
          IF (CHARLY) GO TO 20000
          PAUSE
          GO TO 1000
 999      ACTION=.TRUE.
          GO TO 99999
 20000 CALL EXIT
 C        TO READ IN AN EBCDIC CARD IMAGE OF A // JOB CARD
```

```
CC*
C***********************************************************************
C       IF THE SUPERVISOR CAN BE ENTERRED TO GIVE IT A CARD IMAGE,
CC      REMOVE THE COMMENTS CHARACTERS FROM THE FOLLOWING CARDS-
C***********************************************************************
CC*                                                                   *
C       IF ((AREA(1).EQ.708).AND.(AREA(2).EQ.705)) GO TO 3000
CC*                                                                   *
C THAT CARD REPLACES THE ONE AT STATEMENT 9
CC*                                                                   *
C3000 DO 2001 IN=1,80
CC*                                                                   *
C
C        THIS LOOP CONVERTS THE CARD TO EBCDIC
C
CC*                                                                   *
C       AREA(IN)=LETTER(AREA(IN))
CC*                                                                   *
C2001 AREA(IN)=SHIFTS(AREA(IN),2,8)
CC*                                                                   *
C       DO 2002 IN=1,80,2
CC*                                                                   *
C       THIS LOOP PACKS THE CHARACTERS TWO TO THE HALFWORD
CC*                                                                   *
C
CC*                                                                   *
C       INDEX=(IN/2)+1
CC*                                                                   *
C2002 AREA(INDEX)=SHIFTS(AREA(IN),1,8)+AREA(IN+1)
C
C***********************************************************************
C***********************************************************************
99999 CONTINUE
C
C
C     THAT ENDS THE SECTION THAT LOCKS FOR THE END
C
C***********************************************************************
C
5000  CONTINUE
      IF (ACTION) GO TO 10000
      NREX=NREX+1
C        SET THE COUNTERS TO ZERO,AND THE PHYSICAL RECORD NUMBER TO NREX
      RECD(45)=NREX
      RECD(44)=0
      RECD(43)=0
      DO 3 ILCOL=1,80,2
      IF (AREA(ILCOL).NE.0) RECD(44)=RECD(44)+1
C             PERFORM CHECK FOR ILLEGAL COLUMNS
    3 CONTINUE
      DO 4 NLCL=1,23
      NDEX=NUMCOL(NLCL)
C                CONVERT THE NUMERIC COLUMNS NOW........
      AREA(NDEX)=NUMBER(AREA(NDEX))
      IF (AREA(NDEX).EQ.-1) RECD(43)=RECD(43)+1
C                AND UPDATE THE NUMERIC COLUMN ERROR COUNTER.......
    4 CONTINUE
      DO 5 IQCOL=1,12
C             MOVE THE QUESTION COLUMNS INTO THE CORRECT AREA
```

```
      NDEX=QCOLS(IQCOL)
    5 RECD(17+IQCOL)=AREA(NDEX)
C
C
C
C       AND FORMAT THE COMMON DATA ON THE CARD
      RECD(1)=(AREA(60)*10)+AREA(62)
      RECD(3)=(AREA(66)*10)+AREA(68)
      RECD(4)=(AREA(54)*10)+AREA(56)
      RECD(5)=(AREA(42)*10)+AREA(44)
      RECD(6)=(AREA(48)*10)+AREA(50)
      ID1=LETTER(AREA(72))
      ID2=LETTER(AREA(74))
      ID3=LETTER(AREA(76))
      ID4=LETTER(AREA(78))
      ID5=LETTER(AREA(80))
C        SEE WHAT KIND OF CARDS ARE ENTERRING
      IF (ICODE.EQ.1) GO TO 100
C
C                 IT'S A STUDENT
      RECD( 2) = ((( AREA(2)*10)+AREA(6))*10)+AREA(8)
      RECD(13)=(AREA(24)*10)+AREA(26)
      RECD(14)=(AREA(12)*10)+AREA(14)
      RECD(15)=(AREA(18)*10)+AREA(20)
      RECD(16)=(AREA(30)*10)+AREA(32)
      RECD(17)=(AREA(36)*10)+AREA(38)
      RECD(7)=ID1
      CALL PACK(ID2,ID3,RECD(8))
      CALL PACK(ID4,ID5,RECD(9))
      GO TO 199
C
C       BELOW HERE, IT'S GOTTA BE A HEADER
  100 RECD(2)=0
      RECD(7)=0
      RECD(8)=0
      RECD(9)=0
      CALL PACK(ID1,ID2,RECD(7))
      CALL PACK(ID3,ID4,RECD(8))
      RECD(9)=ID5
      RECD(10)=0
      RECD(12)=0
      RECD(13)=(AREA(24)*10)+AREA(26)
      RECD(11) = ((( AREA(2)*10)+AREA(6))*10)+AREA(8)
      RECD(14)=0
      RECD(15)=0
      RECD(16)=0
      RECD(17)=0
      RECD(10)=LETTER(AREA(80))
C     BUT NOT OVER HERE IT DOESN'T
  199 CALL MERGE(RECD,ICNTRL,FIRST)
  200 CONTINUE
      GO TO 2
10000 CONTINUE
      END FILE ITAPE
      REWIND ITAPE
 2000 CONTINUE
      RETURN
      END
      SUBROUTINE MERGE(RECD,ICNTRL,FIRST)
```

```
C***********************************************************************
C***********************************************************************
C      SUBROUTINE MERGE(RECD,ICNTRL,IVECT,FIRST)
C  THIS SUBROUTINE HANDLES THE MERGING OF INPUT CARDS
C      INTO AIMS TAPE RECORD FORMAT.
C   IT ALSO HANDLES THE SINGLE AND DOUBLE DECISION AND FORMATTING.
C    SUBROUTINE ERRER CHECKS FOR ERRORS IN INCOMING RECORDS.
C   SUBROUTINE DOUBLEAND SUBROUTINE SINGLE HANDLE DOUBLE AND SINGLE
C   RECORD FORMATTING.
C
C***********************************************************************
C***********************************************************************
      DIMENSION ICNTRL(15)
      INTEGER OUTAP
      INTEGER*2 ISAVE
      INTEGER*2 S,D
      INTEGER*2 DE,LE,T
      INTEGER*2 ILES
      INTEGER*2 RECD(45)
      INTEGER*2 IVECT(4,10,40)
      INTEGER*2 CHKRC(45)
      INTEGER*2 OUTPUT(65)
      INTEGER*2 NRMTY,ITAPE,JTYPE,NORM
      INTEGER*2 TNUM
      INTEGER*2 INUM
      COMMON/CHKREC/IVECT
      LOGICAL*1 ACTION
      LOGICAL*1 FIRST
      LOGICAL*1 DELET
      EQUIVALENCE (OUTPUT(12),TNUM)
      DATA DE/'DE'/,LE/'LE'/,T/'T'/
      DATA S/'S'/,D/'D'/
 6000 CONTINUE
      IPT=3
      NORM=ICNTRL(3)
      JTYPE=ICNTRL(4)
      ITAPE=ICNTRL(2)
      NRMTY=ICNTRL(5)
      ILES=RECD(1)
      DELET=.FALSE.
      IF (.NOT.FIRST) GO TO 100
C.......... THIS SECTION IS ONLY EXECUTED ONCE FOR ANY CALL TO CITO
      FIRST=.FAL
      DO 1 IN=1,6
C   .......THAT CLEARS THE OUTPUT RECORD..........
    1 OUTPUT(IN)=0
C........
C   THEN SET UP A FEW REALLY FUN PARAMETERS
C                         ................
      CHKRC(5)=ICNTRL(1)
      ISAVE=CHKRC(5)
      CHKRC(3)=ICNTRL(6)
      CHKRC(2)=ICNTRL(7)
      OUTAP=ICNTRL(8)
C    SET THE SEQUENCE NUMBERS EQUAL TO ZERO
      ISEQ=0
C    AND DROP THROUGH TO AN EXECUTION OF THE OTHER JUNK AND SUCH LIKE
  100 CONTINUE
C    THIS SECTION IS ATTEMPTED FOR EVERY CARD
```

```
      CALL ERROR(RECD,CHKREC,OUTPUT,ISEQ,JTYPE,ACTION)
C          IS THE RECORD LEGAL
      IF (.NOT.ACTION) GO TO 77777
      IF (ISEQ.GT.0) GO TO 2003
      IF ((RECD(7).EQ.DE).AND.(RECD(8).EQ.LE).AND.(RECD(9).EQ.T)) DELET=
     2.TRUE.
      IF (DELET) GO TO 2003
      ICNT=0
C     IF IT ISN'T THE FIRST CARD, I ALREADY KNOW ALL ABOUT IT
C      SEE WHAT KIND OF CARDS THEY ARE
      IF  (JTYPE.EQ.1) GO TO 1000
C     1=HEADER, 2=STUDENT
C
C   THIS NEXT SECTION LOOKS FOR THE CARD IN THE AIMS RECORDS
C...........AND HOPEFULLY FINDS IT
C                               -----
      DO 2 IDECK=1,10
      IF (IVECT(1,IDECK,ILES).NE.RECD(3) ) GO TO 2
      IF (IVECT(2,IDECK,ILES).NE.RECD(4) ) GO TO 2
      GO TO 5
    2 CONTINUE
      CALL ERR(RECD,ISEQ,JTYPE,ICNTRL(1))
      WRITE (IPT,3)
    3 FORMAT (1H ,'REQUESTED LESSON,SEGMENT AND TYPE DON''T EXIST IN AIM
     2S LESSON DATA FILES. PLEASE RE-CHECK.',/1H0,' ABOVE CARD REJECTED'
     3,//)
77777 DO 4 IN=1,65
    4 OUTPUT(IN)=0
      ISEQ=0
      GO TO 99999
    5 ITY=IVECT(3,IDECK,ILES)
      IF (ITY.EQ.0) ITY=S
      IF (ITY.EQ.1) ITY=D
      INUM=IVECT(4,IDECK,ILES)
      GO TO 2000
C
C     THAT'S THE SECTION THAT FINDS OUT ALL ABOUT A STUDENT RECORD
C
C   THIS,ON THE OTHER HAND, DISCOVERS WHAT GIVES ABOUT A HEADER CARD
 1000 CONTINUE
C     THIS SECTION MAKES SURE I HAVE DATA ON A HEADER CARD
      IF((RECD(10).NE.S).AND.(RECD(10).NE.D))
     2 CALL ERR(RECD,ISEQ,JTYPE,ICNTRL(1))
      IF((RECD(10).NE.S).AND.(RECD(10).NE.D))WRITE (IPT,1001)
      IF((RECD(10).NE.S).AND.(RECD(10).NE.D)) GO TO 77777
 1001 FORMAT (1H ,' NO  SINGLE OR DOUBLE INFORMATION ON HEADER CARD',/1H
     20,'ABOVE CARD REJECTED.',//)
      ITY=RECD(10)
      INUM=0
      IF ((RECD(10).EQ.S).AND.(RECD(13).LE.48)) INUM=2
      IF ((RECD(10).EQ.S).AND.(RECD(13).LE.24)) INUM=1
      IF ((RECD(10).EQ.D).AND.(RECD(13).LE.48)) INUM=4
      IF ((RECD(10).EQ.D).AND.(RECD(13).LE.36)) INUM=3
      IF ((RECD(10).EQ.D).AND.(RECD(13).LE.24)) INUM=2
      IF ((RECD(10).EQ.D).AND.(RECD(13).LE.12)) INUM=1
      IF(INUM.GT.1) RECD(6)=INUM
      IF (INUM.NE.0) GO TO 2000
C............THAT'S IT
C DEAR D. S. YOU HAVE MADE A NONO. THE PREVIOUS STATEMENT WAS INSERTED
```

```
        AIPS III SOURCE STATEMENT LISTING

C TO CORRECT THE FACT  THAT THERE WAS NO WAY OF TO UPDATING RECD(6)
C  FOR A MULTI-CARD HEADER . IT WOULD BE CONATSNTLY O.K.      D. L.
      CALL ERR(RECD,ISEQ,JTYPE,ICNTRL(1))
      WRITE (IPT,1004)
 1004 FORMAT (1H ,'INVALID OR MISSING DATA FOR THE NUMBER OF RECORDS',/1
     2H0,'ABOVE RECORD REJECTED',//)
      GO TO 77777
C THIS NEXT SECTION SEES WHAT TO DO NEXT
C
 2000 CONTINUE
      TNUM=0
      DO 2001 IN=1,17
 2001 OUTPUT(IN)=RECD(IN)
      ISTART=18
      IF (ITY.EQ.S) INCRE=24
      IF (ITY.EQ.D) INCRE=12
      DO 2002 IN=18,65
 2002 OUTPUT(IN)=0
 2003 ICNT=ICNT+1
      IF (DELET) INUM=ICNT
      IF (DELET) GO TO 2006
      IF (ITY.EQ.S) CALL SINGLE(RECD(18)        ,NORM,NRMTY,RECD(4),NUM)
      IF (ITY.EQ.D) CALL DOUBLE(RECD(18)        ,NUM)
      ISTART=18+((ICNT-1)*INCRE)
      IEND=ISTART+INCRE-1
      IF (NUM.EQ.-1) CALL ERR(RECD,ISEQ,JTYPE,ICNTRL(1))
      IF (NUM.EQ.-1) WRITE(IPT,2005)
 2005 FORMAT (1H ,'ERRORS IN QUESTION COLUMNS.',/1H ,' ABOVE CARD REJECT
     2ED',//)
      IF (NUM.EQ.-1) GO TO 77777
      DO 2004 IN=ISTART,IEND
      ION=IN-((ICNT-1)*INCRE)
 2004 OUTPUT(IN)=RECD(ION)
      TNUM=TNUM+NUM
      ISEQ=ISEQ+1
 2006 IF (ICNT.LT.INUM) RETURN
      OUTPUT(12)=TNUM
      IF (ITY.EQ.S) OUTPUT(10)=0
      IF (ITY.EQ.D) OUTPUT(10)=1
      IF (ICNTRL(2).NE.2) GO TO 77777
 6002 CONTINUE
      WRITE (OUTAP) (OUTPUT(INK),INK=1,65)
      WRITE(3,6003)(OUTPUT(IDUM),IDUM=1,65)
 6003 FORMAT(4(20I6,/))
      GO TO 77777
99999 CONTINUE
      CHKRC(5)=ISAVE
 6001 CONTINUE
      RETURN
      END
      SUBROUTINE ERRER(RECD,CHKRC,AREA,SEQ,JTYPE,ACTION)
C     SUBROUTINE ERRER
C     THIS PROGRAM CHECKS FOR ERRORS IN RECORDS
C     BY DENNIS I. SCHNEIDER
C**********************************************************************
      INTEGER SEQ
      INTEGER*2 ICRES
      INTEGER*2 JTYPE
      INTEGER*2 RECD(45),CHKRC(45),AREA(65)            ,INFORM(12)
```

AIMS III SOURCE STATEMENT LISTING

```
      INTEGER*2 DE,LE,T
      INTEGER*2 SP
      LOGICAL*1 CONTRL(45),CNTRL(45)
      LOGICAL*1 DELET
      LOGICAL*1 INTC,ILCOL,ACTION,P
      LOGICAL*1 FALSE,TRUE
      LOGICAL*1 NUMS
      EQUIVALENCE(CNTRL(1),CONTRL(1))
      COMMON /SYSTEM/NLESS,NDECK,NREX,NQUEST,NSTU
      DATA SP/' '/
      DATA DE/'DE'/,LE/'LE'/,T/'T'/
      IERR=0
      MIDDLE=0
      ICRES=CHKRC(5)
      IPT=3
      NUMS=.FALSE.
      FALSE=.FALSE.
      TRUE=.TRUE.
      P=.FALSE.
      ILCOL=.FALSE.
      INTC=.FALSE.
      DELET=.FALSE.
      DO 1 K=1,45
    1 CONTRL(K)=.FALSE.
      IF (RECD(43).NE.0) NUMS=.TRUE.
      IF (RECD(1).LE.0) CONTRL(1)=.TRUE.
      IF (RECD(1).GT.NLESS) CONTRL(1)=.TRUE.
      IF (JTYPE.EQ.1) GO TO 2
C           JTYPE=1 FOR HEADER,2 FOR STUDENT
      IF (RECD(2).GT.NSTUD) CONTRL(2)=.TRUE.
      IF (RECD(2).LE.0) CONTRL(2)=.TRUE.
    2 IF (RECD(3).LE.CHKRC(3)) CONTRL(3)=.TRUE.
C                    CHKRC(3)IS LOWER BOUND FOR SEGMENT NUMBER
C                    CHKRC(2) IS UPPER BOUND FOR SEGMENT NUMBER
      IF (RECD(4).LE.0) CONTRL(4)=.TRUE.
      IF (RECD(3).GT.CHKRC(2)) CONTRL(3)=.TRUE.
      IF (RECD(5).NE.CHKRC(5)) CONTRL(5)=.TRUE.
      IF (RECD(6).NE.(SEQ+1))  CONTRL(6)=.TRUE.
      IF ((RECD(7).EQ.DE).AND.(RECD(8).EQ.LE).AND.(RECD(9).EQ.T)) DELET=
     2.TRUE.
      IF (DELET) GO TO 4
      IF (JTYPE.NE.2) GO TO 3
      GO TO 4
    3 IF ((RECD(7)+RECD(8)+RECD(9)).EQ.0) CONTRL(9)=.TRUE.
      IF ((JTYPE.EQ.1).AND.(RECD(10).EQ.SP)) CONTRL(10)=.TRUE.
      IF ((JTYPE.EQ.1).AND.((RECD(11).LE.0).OR.(RECD(11).GT.(24*5*NREX))
     2))CONTRL(11)=.TRUE.
      IF ((JTYPE.EQ.1).AND.((RECD(13).LE.0).OR.(RECD(13).GT.(24*NREX)))
     2)CONTRL(13)=.TRUE.
    4 IF (RECD(44).NE.0) ILCOL=.TRUE.
      IF (SEQ.EQ.0) GO TO 100
C                          SECTION BELOW MAKES SURE OF I.D. AREA MATCH
      IF (JTYPE.EQ.1) INDEX=10
      IF (JTYPE.EQ.2) INDEX=9
      DO 5 I=1,5
      IF (RECD(I).NE.AREA(I))  IERR=IERR+1
    5 CONTINUE
      DO 6 I=7,INDEX
      IF (RECD(I).NE.AREA(I))  IERR=IERR+1
      RIMS III SOURCE STATEMENT LISTING

    6 CONTINUE
      IF (IERR.EQ.0)IMTC=.FALSE.
C                         BELOW HERE,ALL IS OUTPUT OR NO OUTPUT
  100 DO 101 I=1,45
      IF (.NOT.CONTRL(I)) GO TO 101
      GO TO 102
  101 CONTINUE
      IF (ILCCL) GO TO 102
      IF (IMTC) GO TO 102
      IF (NUMS) GO TO 102
      GO TO 7000
      ENTRY ERR(RECD, SEQ,JTYPE,ICRES)
      MIDDLE=1
  102 CALL INFO(INFORM)
      WRITE (IPT,103) INFORM,RECD(45),SEQ,RECD(6),JTYPE
  103 FORMAT (1H0,/1H ,'**** CARD IN ERROR **** JOB NAME ',4A2,5X,'DATE-
     2 ',4A2,5X,'TIME- ',4A2,'  **********PHYSICAL RECORD - ',I8,/1H ,
     3'PREVIOUS SEQUENCE NO.= ',I2,' CARD SEQUENCE NO.= ',I2,'  TYPE OF
     4 CARDS = ',I1,' (1=HEADER,2=STUDENT     ) ' )
      WRITE (IPT,117) ICRES   ,RECD(5),RECD(1),RECD(3),RECD(4)
  117 FORMAT (1H ,'COURSE BEING PROCESSED = ',I2,' COURSE NUMBER ON CARD
     2 = ',I2,' LESSON = ',I2,' SEGMENT = ',I2,' TYPE = ',I2)
      IF (JTYPE.EQ.1) WRITE (IPT,118) RECD(13),RECD(11),(RECD(N),N=7,9),
     2RECD(10)
  118 FORMAT (1H ,'NUMBER OF QUESTIONS = ',I2,' NUMBER OF SELECTIONS = '
     2,I3,' I.D.FIELD = ',3A2,' (S) OR (D) = 'A1)
      IF (JTYPE.EQ.2) WRITE (IPT,119) RECD(2),(RECD(N),N=7,9),RECD(14),
     2RECD(15),RECD(13),RECD(16),RECD(17)
  119 FORMAT (1H ,'COURSE STUDENT NUMBER = ',I3,' STUDENT I.D. NUMBER =
     2 '3A2,' DATE ON CARD = ',I2,'/',I2,'/',I2,' TIME ON CARD = ',I2,
     3'.',I2)
      WRITE (IPT,121)
  121 FORMAT (1H ,'------------------ E R R O R S  -------------------
     2-------------------------------------------------------',/)
      IF (MIDDLE.EQ.1) RETURN
      IF (DELET) WRITE (IPT,116)
  116 FORMAT (1H ,'THIS RECORD HAS BEEN FOUND TO BE A DLETE RECORD')
      IF (CONTRL(1)) WRITE (IPT,104)
  104 FORMAT (1H ,'LESSON NUMBER MISSING,UNINTELLIGIBLE OR MIS-MATCH.')
      IF (CONTRL(2)) WRITE (IPT,105)
  105 FORMAT (1H ,'COURSE STUDENT NUMBER MISSING,UNRECOGNIZABLE OR EXCEE
     2DS SYSTEM LIMITATION NSTUD.')
      IF (CONTRL(3)) WRITE (IPT,106)
  106 FORMAT (1H ,'SEGMENT NUMBER MISSING,MISPUNCHED OR UNRECOGNIZABLE,O
     2R PREVIOUS CARDS IN RECORD BAD.')
      IF (CONTRL(4)) WRITE (IPT,107)
  107 FORMAT (1H ,'CARD TYPE NUMBER MISSING OR UNRECOGNIZABLE.')
      IF (CONTRL(5)) WRITE (IPT,108)
  108 FORMAT (1H ,'COURSE NUMBER NOT AS PER SYSTEM SPECIFICATIONS')
      IF (CONTRL(6)) WRITE (IPT,109)
  109 FORMAT(1H ,'SEQUENCE NUMBER MISSING OR BAD, OR PREVIOUS CARD(S) I
     2N THIS RECORD BAD.')
      IF (CNTRL(9).AND.(JTYPE.EQ.2)) WRITE (IPT,110)
  110 FORMAT (1H ,'DATA IN I.D. NUMBER AREA BAD OR MISSING-STUDENT RUN')
      IF (CNTRL(10).AND.(JTYPE.EQ.1)) WRITE (IPT,111)
  111 FORMAT (1H ,'DATA IN I.D. NUMBER AREA BAD OR MISSING-HEADER RUN.')
      IF (CNTRL(11)) WRITE (IPT,112)
  112 FORMAT (1H ,'NUMBER OF SELECTIONS NOT AS PER SPECIFICATION.')
      IF (CNTRL(13)) WRITE (IPT,113)
  113 FORMAT (1H ,'NUMBER OF QUESTIONS  NOT AS PER SPECIFICATIONS.')
      IF (ILCOL) WRITE (IPT,114) RECD(44)
  114 FORMAT (1H ,'ILLEGAL USE OF ODD-NUMBERED COLUMNS-',I4,'-TIME(S).')
      IF (IMTC)  WRITE (IPT,115) IERR
  115 FORMAT (1H ,I5,' OCCASIONS IN WHICH CARD''S IDENTIFICATION DATA IS
     2 MIS-MATCHED WITH THE REMAINDER OF THIS RECORD.')
      IF (NUMS) WRITE (IPT,120)         RECD(43)
  120 FORMAT (1H ,'MISPUNCHED NUMERIC COLUMNS FOUND - ',I3, ' TIMES.',/)
      ACTION=.FALSE.
      GO TO 7001
 7000 ACTION=TRUE
 7001 IF ((JTYPE.EQ.2).AND.((RECD(13).LT.0).OR.(RECD(13).GT.99)))P=.TRUE
     2.
      IF ((JTYPE.EQ.2).AND.((RECD(14).LT.0).OR.(RECD(13).GT.12)))P=.TRUE
     2.
      IF ((JTYPE.EQ.2).AND.((RECD(15).LT.0).OR.(RECD(15).GT.31)))P=.TRUE
     2.
      IF ((JTYPE.EQ.2).AND.((RECD(16).LT.0).OR.(RECD(16).GT.99)))P=.TRUE
     2.
      IF ((JTYPE.EQ.2).AND.((RECD(17).LT.0).OR.(RECD(16).GT.59)))P=.TRUE
     2.
      IF (.NOT.P) GO TO 8000
      IF (ACTION) CALL INFO (INFORM)
      IF (ACTION) WRITE (IPT,103) INFORM,RECD(45),SEQ,RECD(6),JTYPE
      IF (P) WRITE (IPT,7999)
 7999 FORMAT (1H ,'ERRORS IN YEAR,MONTH,DAY,HOURS OR MINUTES COLUMNS - E
C
C     THIS ROUTINE PERFORMS ONE KIND OF DOUBLE CONVERSION
     2RRORS IGNORED.')
 8000 IF (ACTION.AND.P) WRITE (IPT,8001)
 8001 FORMAT (1H0,'**** ABOVE CARD ACCEPTED DESPITE ERRORS. *****',////)
      IF (.NOT.ACTION) WRITE (IPT,8002)
 8002 FORMAT (1H0,'**** ABOVE CARD HAS BEEN REJECTED **** PLEASE NOTE TH
     2AT IN A MULTI-CARD RECORD,THIS WILL CAUSE OTHER ERRORS **',//////)
      IF (ACTION)  GO TO 10000
      DO 8003 IN=1,65
 8003 AREA(IN)=0
      SEQ=0
10000 RETURN
      END
      SUBROUTINE DOUBLE(AREA,NUM)
C    AS FAR AS I'M CONCERNED, TO 45/  WITH DIFFERENT KINDS OF CARDS
C     BY DENNIS I. SCHNEIDER
      INTEGER*2 SHIFTS
      INTEGER*2 AREA(12)
      NUM=0
      DO 200 IER=1,12
      IF (SHIFTS(AREA(IER),RIGHT,10).NE.0) NUM=-1
      IF (NUM.EQ.-1) GO TO 999
200   CONTINUE
      DO 100 IQUES=1,12
      AREA(IQUES)=SHIFTS(AREA(IQUES),LEFT,1)+1
      IF (AREA(IQUES).NE.1) NUM=NUM+1
100   CONTINUE
999   RETURN
      END
      SUBROUTINE SINGLE(AREA,NORM,NRMTY,ITYPE,NUM)
C    THIS ROUTINE HANDLES SINGLE CONVERSION
C     BY DENNIS I. SCHNEIDER
```

```
 ATPS III SOURCE STATEMENT LISTING

      INTEGER RIGHT,LEFT
      INTEGER*2  NORM,NRMTY,ITYPE
      INTEGER*2 SHIFTS
      INTEGER*2 AREA(24)
      INTEGER*2 SAVE(24)
      INTEGER*2 HICDE,LWCDE
      INTEGER*2 CODE
      LEFT=1
      RIGHT=2
      IPT=3
      NUM=0
      DO 999 IN=1,24
  999 SAVE(IN)=0
      DO 10000 IQUES=1,12
      CODE=AREA(IQUES)
      IF (NORM.EQ.5) GO TO 100
      IF (NORM.EQ.4) GO TO 200
      WRITE (IPT,1)
1     FORMAT (1H ,'ILLEGAL NORMAL CARD TYPE CODE- SINGLE.',/)
      CALL EXIT
  100 IF (NRMTY.EQ.ITYPE)       GO TO 2000
      GO TO 1000
C        THIS IS THE END OF THE FIVE-CODE NORMAL SECTION
  200 IF (NRMTY.EQ.ITYPE)       GO TO 1000
      GO TO  2000
C        THIS IS THE END OF THE FOUR-CODE NORMAL SECTION
C       A BRANCH TO 1000 INDICATES 5-CODE FORMATTING
C     A BRANCH TO 2000 INDICATES  4-CODE FORMATTING
C*********************************************************************
C*********************************************************************
1000  CONTINUE
C
C         THESE ARE THE SHIFTS TO FORM THE CODE FOR FIVE RESPONSES
C
      HICDE=SHIFTS(CODE,RIGHT,5)
      LWCDE=CODE-SHIFTS(HICDE,LEFT,5)
      HICDE=SHIFTS(HICDE,LEFT,1)+1
      LWCDE=SHIFTS(LWCDE,LEFT,1)+1
C
C*********************************************************************
      GO TO 3000
C*********************************************************************
C
C
C*********************************************************************
C*********************************************************************
 2000 CONTINUE
C
C          THESE ARE THE SHIFTS TO FORM CODES FOR FOUR RESPONSES
C
      HICDE=SHIFTS(CODE,RIGHT,5)
      LWCDE=CODE-SHIFTS(HICDE,LEFT,5)
      HICDE=SHIFTS(HICDE,RIGHT,1)
      LWCDE=SHIFTS(LWCDE,RIGHT,1)
      HICDE=SHIFTS(HICDE,LEFT,1)+1
      LWCDE=SHIFTS(LWCDE,LEFT,1)+1
C
C*********************************************************************
      GO TO 3000
```

AIMS III SOURCE STATEMENT LISTING

```
C*****************************************************************
 5000 CONTINUE
      SAVE(2*IQUES-1)=LOCODE
      SAVE(2*IQUES)=HICODE
10000 CONTINUE
      DO 997 IERR=1,12
      IF (SHIFTS(AREA(IERR),RIGHT,10).NE.0) NUM=-1
      IF (NUM.EQ.-1)          GO TO 998
997   CONTINUE
      DO 998 IN=1,24
      IF (SAVE(IN).GT.1) NUM=NUM+1
      AREA(IN)=SAVE(IN)
998   CONTINUE
      RETURN
      END
      FUNCTION NUMBER(ICODE)
C  THIS ROUTINE HAS BEEN CATALOGED IN THE RELOCATABLE LIBRARY
      INTEGER*2 ICODE,NUMBER
      INTEGER*2 IESTR(10)
      DATA IESTR/512,256,128,64,32,16,8,4,2,1/
      IF (ICODE.EQ.0) NUMBER=NUMBER-NUMBER
C  IS IT AN EXCEPTION
      IF (ICODE.EQ.0) GO TO 7000
C     A DROP THRU MEANS NO
      DO 2 N=1,10
      IF (ICODE.EQ.IESTR(N)) GO TO 3
C    .......LOK IT UP IN THE TABLE TO SEE IF IT'S THERE AT ALL
    2 CONTINUE
      NUMBER=-1
C     A DROP THRU MEANS IT'S ABSENT
      GO TO 7000
    3 NUMBER =N-1
C      THIS SECTION FINDS THE NUMERIC EQUIVALENT AND SPLITS
 7000 RETURN
      END
      FUNCTION LETTER(ICODE)
C  THIS ROUTINE HAS BEEN CATALOGED IN THE RELOCATABLE LIBRARY
C    THIS ROUTINE PERFORMS EBCDIC CONVERSIONS
C     BY DENNIS I. SCHNEIDER
      INTEGER*2 ICODE,LETTER,EQUIV(40),EBCDIC(40)
      INTEGER*2 IAST,ISP
      DATA EBCDIC/'A','B','C','D','E','F','G','H','I','J','K','L','M',
     2'N','O','P','Q','R','S','T','U','V','W','X','Y','Z'
     3,'0','1','2','3','4','5','6','7','8','9',
     4'/','*','(','&'/
      DATA EQUIV/2304,2176,2112,2080,2064,2056,2052,2050,
     22049,1280,1152,1088,1056,1040,1032,1028,
     31026,1025,640,576,544,528,520,516,
     4514,513,512,256,128,64,32,16,
     58,4,2,1,768,1056,2066,2048/
      DATA IAST/'*'/,ISP/' '/
C    FIRST,SEE IF IT'S AN EXCEPTION
      IF (ICODE.EQ.0) LETTER=ISP
      IF (ICODE.EQ.0) GO TO 999
      DO 1 N=1,40
C    .....NEXT,IS IT IN THE TABLE.......
      IF (ICODE.EQ.EQUIV(N)) GO TO 100
    1 CONTINUE
C    A DROP THRU TO HERE MEANS IT'S ABSENT FROM THE TABLE
```

FILMED FROM BEST AVAILABLE COPY

```
        AIMS III SOURCE STATEMENT LISTING

       LETTER = LAST
       GO TO 999
   100 LETTER=EBCDIC(N)
 C       THIS PART JUST FETCHES THE EQUIVALENT AND SPLITS
   999 RETURN
       END
 /*
 // ASSGN SYSSLB,X'191'
 // EXEC ASSEMBLY
 INFO       START
            COME ENTER
            PRINT NOGEN
 JOB        DS    2F
 DATE       DS    2F
 TIME       DS    2F
 PAR        DS    1F
            DS    3C
 PAT        DC    10X'F0202021482020482020'
 WORK       DS    10C
 INTIME     DS    1F
 ENTER      L     7,0(0,1)
            GETIME STANDARD
            ST    1,INTIME
            MVC   WORK(10),PAT
            ED    WORK(10),INTIME
            MVC   TIME(8),WORK+2
            COMRG
            MVC   DATE(8),0(1)
            MVC   JOB(8),23(1)
            MVC   0(24,7),JOB
            PRINT GEN
            HOME
            END
 /*
 // EXEC ASSEMBLY
 PACK       START 0
            COME  ENTER
 A          DS    1H
 B          DS    1H
 LOCALE     DSECT
 FIELD      DS    1H
 PACK       CSECT
 ENTER      L     3,0(0,1)
            LH    4,0(0,3)
            STH   4,A
            L     3,4(0,1)
            LH    4,0(0,3)
            STH   4,B
            L     3,8(0,1)
            USING LOCALE,3
            LA    4,FIELD
            MVC   0(1,4),A
            MVC   1(1,4),B
            HOME
            END
 /*
 // EXEC ASSEMBLY
 CRDIN      START 0
 *  THIS ROUTINE HAS BEEN CATALOGED IN THE RELOCATABLE LIBRARY
```

AIMS III SOURCE STATEMENT LISTING

```
*     ENTRY MACRO BRANCHES TO 'ENTER'
          COME  ENTER
*
CCB1      CCB   SYS015,WORK
*
*       PHYSICAL IOCS SECTION
*
WORK      CCW   X'22',AREA1,X'00',160
*
AREA1     DS    80H
SAVER     DS    9D
*
ENTER     STM   0,15,SAVER
          EXCP  CCB1
* GET IN A CARD FROM THE READER
          WAIT  CCB1
          L     5,=F'0080'
*     THIS NEXT SECTION RIGHT - JUSTIFIES EACH COLUMN IN A HALFWORD
LOOP      LR    6,5
          S     6,=F'0001'
          MH    6,=H'0002'
          LH    8,AREA1(6)
          SRDA  8,6(0)
          SRA   8,2(0)
          SLDA  8,6(0)
          N     8,=X'00000FFF'
          STH   8,AREA1(6)
          BCT   5,LOOP
          LM    0,15,SAVER
          L     4,0(0,1)
          LA    5,AREA1
*     PLACE THE DATA IN THE PARAMETER AREA
          MVC   0(160,4),0(5)
          LM    0,15,SAVER
*............... AND SPLIT
          HOME
          EOJ
          LTORG
          END
/*
// EXEC ASSEMBLY
SHIFTS    START 0
*  THIS ROUTINE HAS BEEN CATALOGED IN THE RELOCATABLE LIBRARY
*   THIS ROUTINE HANDLES ALL SHIFTING OPERATIONS FOR FORTRAN
*         ENTRY MACRO CAUSES BRANCH TO 'ENTER'
          COME  ENTER
*
*
*      PARAMETER STORAGE AREA
*
*
SAVER     DS    9D
DIRECTN   DS    1F
DISTNCE   DS    1F
DATA      DS    1H
*
*      END OF PARAMETER STORAGE AREA
*
ENTER     STM   0,15,SAVER
```

```
       KIPS III SOURCE STATEMENT LISTING

*  THAT SAVES ALL THE GPR'S,JUST IN CASE.......
*
*
*   THIS SECTION FETCHES THE ENTRY PARAMETERS
*
             L      4,0(0,1)
             LH     5,0(0,4)
             STH    5,DATA
             L      4,4(0,1)
             L      5,0(0,4)
             ST     5,DIRECTN
             L      4,8(0,1)
             L      5,0(0,4)
             ST     5,DISTNCE
*    I NOW HAVE ALL THE PARAMETERS
*
*     NEXT, I RELOAD THE DATA.........
*
             LH     8,DATA
             L      5,DISTNCE
             L      4,DIRECTN
*
*     .........AND THEN..........
*
             C      4,=F'0001'
*
*         IT'S TEST-THE-CODE TIME,FOLKS.......
*              CODE = 1 (LEFT) , 2 (RIGHT)
*
             BE     LEFT
*
*  THIS IS THE RIGHT-SHIFT SECTION
*
             SRL    8,0(5)
             B      SPLIT
*
*
*   .....AND THIS ,WANDERING STRANGER,IS THE LEFT SHIFT SECTION
*
LEFT         SLL    8,0(5)
SPLIT        STH    8,DATA
             LM     0,15,SAVER
             LH     0,DATA
* EXIT MACRO ALLOWS RETURN TO FORTRAN
             HOME
             EOJ
             LTORG
             END
/*
// LBLTYP NSD(5)
// EXEC LNKEDT
/*
/&
// JOB 304138        AHEMMAIN
// ASSGN SYSLNK,X'192'
// DLBL IJSYSLN,'SYSLNK',66/365,SD
// EXTENT SYSLNK,000002,1,0,10,1000
// OPTION CATAL
 PHASE ALASMAIN,ROOT
```

AIMS III SOURCE STATEMENT LISTING

```
// EXEC FFORTRAN
C* AIMS MAIN
      COMMON/SYSTEM/NLESS,NDECK,NREX,NQUEST,NSTUD
         COMMON/FILES/IFIL(15)
         INTEGER DATES(6)
         INTEGER SEGMNT,CONTRO(14),PNAME(2)
         INTEGER * 2 COLUMN(40) ,IA   ,  IDENT
         EQUIVALENCE (COLUMN(1),IA),(CARD(1),IB),
     2   (CARD(2),IC),(CARD(3),ID),(CARD(4),IE)
         INTEGER CARD(4)
         DATA SEGMNT/0/,IDENT/'(('/,
     2   CONTRO/'HEAD','STUD','QUES','PROC','HECK','TERF','RINT','-INF',
     3   '-TO-','-LIS','INPU','DROP','REST','LIST'/
         DATA PNAME/'MONI','TOR'/
      DEFINE FILE 5(1500,35,U,I5),
     2             6( 800,33,U,I6),
     3             7( 200,100,U,I7),
     4             8(4000,35,U,I8),
     5             9( 250,33,U,I9),
     6            10(8040,35,U,I10),
     7            11( 500,25,U,I11),
     8            12( 100,25,U,I12)
      NLESS = 40
      NDECK = 10
      NREX = 2
      NQUEST = 43
      NSTUD = 185
      DO 7 I = 1,15
    7 IFIL(I) = I
         IN=1
         IOUT = 3
  100    CONTINUE
  101    FORMAT(40A2)
  102    FORMAT(1H1,6X,'***  AIMS  ***'      68X,'JOB'3X,2A4,1X,2A4,1X,
     2   2A4/   1H0,6X,40A2)
         READ(IN,101,END=9000) COLUMN
         IF(IA.NE.IDENT) GO TO 100
         CALL INFO(DATES)
         WRITE(IOUT,102) DATES,COLUMN
         CALL IMAGE(COLUMN,CARD)
         IF(IB.NE.CONTRO(1)) GO TO 2000
 1000    CONTINUE
C A HEADER FUNCTION, GET THE SEGMENT IF NOT IN CORE
         IF(SEGMNT.EQ.1) GO TO 1500
         CALL CPSYS('LOAD','AIMSHEAD')
         SEGMNT = 1
 1500    CONTINUE
         IF(IC.NE.CONTRO(5) ) GO TO 1700
         CALL HEADER
         GO TO 100
 1700    CONTINUE
         IF(IC.NE.CONTRO(6) ) GO TO 1900
C STORE TEMPORARY IN PERMANENT
         CALL STORE
         GO TO 100
 1900    CONTINUE
         IF(IC.NE.CONTRO(7) ) GO TO 7000
         CALL HDLIST
         GO TO 100
 2000 CONTINUE
      IF(IB.NE.CONTRO(2)) GO TO 3000
      IF(SEGMNT.EQ.2)GO TO 2500
      CALL OPSYS('LOAD','AIMSCLAS')
      SEGMNT = 2
 2500 CONTINUE
      IF(IC.NE.CONTRO(13)) GO TO 2700
C  THIS IS FOR A COURSE ROSTER
      CALL LIST
      GO TO 100
 2700 CONTINUE
      IF(ID.EQ.0) ID = 1
      IF(ID.EQ.1.OR.ID.EQ.13) GO TO 2800
      CALL ERROR(PNAME,0,2,ID)
      GO TO 100
 2800 CONTINUE
      IF(IC.NE.CONTRO(11)) GO TO 2900
      CALL INPUT(ID)
      GO TO 100
 2900 CONTINUE
      IF(IC.NE.CONTRO(12)) GO TO 7000
      CALL DROP(ID)
      GO TO 100
 3000 CONTINUE
      IF(IB.NE.CONTRO(3) ) GO TO 4000
C THIS IS A TC FUNCTION
      IF(SEGMNT.EQ.3) GO TO 3500
      CALL OPSYS('LOAD','AIMSQUST')
 3500 CONTINUE
      IF(IC.NE.CONTRO(8) ) GO TO 3700
C ADD + CHAIN MBO INFORMATION
      IF(ID.EQ.0) ID = 1
      IF(ID.EQ.1.OR.ID.EQ.13) GO TO 3600
      CALL ERROR(PNAME,0,2,ID)
      GO TO 100
 3600 CALL MBO1(ID,IE)
      GO TO 100
 3700 CONTINUE
      IF(IC.NE.CONTRO(9) ) GO TO 3900
C LIST THE TO FILE
      CALL MBOLST
      GO TO 100
 3900 CONTINUE
      IF(IC.NE.CONTRO(10) ) GO TO 7000
C LIST THE QUESTION FILE
      CALL QLIST
      GO TO 100
 4000 CONTINUE
      IF(IB.NE.CONTRO(4) ) GO TO 5000
C ITS TO GRADE A LESSON
      CALL OPSYS('LOAD','AIMSPRC1')
      SEGMNT = 4
      CALL PROCES(ID)
      GO TO 100
 5000 CONTINUE
      IF(IB.NE.CONTRO(14) ) GO TO 7000
      IF(SEGMNT.EQ.5) GO TO 5500
      SEGMNT = 5
      CALL OPSYS('LOAD','AIMSLIST')
```

AIMS III SOURCE STATEMENT LISTING

```
  5500     CONTINUE
           CALL RLIST(ID)
           GO TO 100
  7000     CONTINUE
 C BAD     CONTROL  CARD
           CALL ERROR(PNAME,0,1)
           GO TO 100
  9000     CONTINUE
           CALL FINISH
           CALL EXIT
           END
 /*
 // ASSGN SYSSLB,X'191'
 // EXEC ASSEMBLY
 IMAGE     START
           COME  ENTER
 WORK      DC    1D'0'
 ENTER     L     3,0(0,1)
           L     4,4(0,1)
           MVC   0(4,4),2(3)
           MVC   4(4,4),10(3)
           CLC   23(2,3),=CL2'  '
           BE    BLANK1
           PACK  WORK+4(4),23(2,3)
           CVB   5,WORK
           BC    X'F',ST1
 BLANK1    SR    5,5
 ST1       ST    5,8(0,4)
           CLI   29(3),C' '
           BE    BLANK2
           PACK  WORK+4(4),29(1,3)
           CVB   5,WORK
           BC    X'F',ST2
 BLANK2    SR    5,5
 ST2       ST    5,12(0,4)
           HOME
           END
 /*
 // EXEC ASSEMBLY
 ERROR     START
           ENTRY FINISH
           COME  ENTER
           PRINT NOGEN
 ENTER     ST    1,PARLST               SAVE THE ADDRESS OF PARAMETERS
           BC    X'0',LATER             BRANCH PAST OPEN ON 2ND ENTRY
           MVI   ENTER+5,X'F0'          DISABLE OPEN AT MAIN ENTRY
           MVI   F+1,X'F0'             DISABLE OPEN AT SECONDARY ENTRY
           OPENR PRINTS                 OPEN FILE
 LATER     L     3,PARLST               GET PARAMETER LIST
           L     4,0(0,3)               GET 1ST PARAMETER ADDR
           MVC   NAME(6),0(4)           1ST PAR IS NAME,MOVED TO PRINT LINE
           L     4,4(0,3)               2ND PARAMETER ADDRESS
           L     5,0(0,4)               GET 2ND PARAMETER
           CVD   5,PDEC                 MAKE IT DECIMAL
           MVC   LEVL(2),PATL           GET A PATTERN
           ED    LEVL(2),PDEC+7         EDIT
           MVC   LEVEL(1),LEVL+1        PLACE IN PRINT LINE
 *                                      IF LEVEL IS POSITIVE WE WILL ABORT
           BC    X'8',A1                TEST IF EDIT RESULT IS 0
```

```
         MVI   BRANCH+1,X'F0'       ENABLE THE BRANCH TO ABORT
         BC    X'F',A2
A1       MVI   BRANCH+1,X'00'       DISABLE THE BRANCH TO ABORT
A2       LA    7,5                  GPR7=5 REGISTER FOR BCT
         LR    8,7                  GPR8=5 COMPARE REGISTER FOR 1ST TIME
         A     3,=F'4'              SET UP GPR3 FOR PARAMETERS
         LA    6,NUM-7              LOAD GPR6 WITH BEGINNING OF LINE
         MVC   NUM+1(34),NUM        BLANK THE PRINT LINE
LOOP     A     3,=F'4'              INDEX THE PARAMETER LIST
         L     4,0(0,3)             GET THE PARAMETER ADDRESS
         L     5,0(0,4)             GET THE DATA
         CVD   5,PDEC               MAKE IT DECIMAL
         A     6,=F'7'              MOVE ALONG OUTPUT LINE
         MVC   0(7,6),PATE          PLACE PATTERN IN OUTPUT LINE
         LA    1,6(0,6)             SIG DIGIT ADDR IN GPR1
         EDMK  0(7,6),PDEC+5        SET UP THE LINE
         BC    X'A',CHECK           IF POSITIVE COMPLETE LOOP
         S     1,=F'1'              GET ADDR FOR SIGN
         MVI   0(1),C'-'            INSERT SIGN IN LINE
         CR    7,8                 SEE IF FIRST PARAMETER(ERROR NO.)
         BC    X'7',CHECK           IF NOT COMPLETE LOOP
         MVI   FLAG+3,X'01'         SET FLAG FOR DUMP
CHECK    TM    0(3),X'FF'           SEE IF LAST PARAMETER
         BC    X'7',WRITE           IF LAST PARAMETER LEAVE
         BCT   7,LOOP               MAXIMUM OF 5 PARAMETERS
WRITE    CALL  INFO,(DATA)         GET JOB NAME,DATE + TIME
         MVC   NAMJ(8),DATA         PLACE NAME IN PRINT LINE
         MVC   DATE(8),DATA+8       PLACE DATE IN PRINT LINE
         MVC   TIME(8),DATA+16      PLACE TIME IN PRINT LINE
         CNTRL PRINTS,SP,3,1
         PUT   PRINTS               WRITE THE LINE
BRANCH   BC    X'1',ABORT           IF LEVEL=1,THIS BRANCHES TO ABORT
         L     1,PARLST             RESTORE GPR1
         PRINT GEN
         HOME                       RETURN TO FORTRAN
         PRINT NOGEN
ABORT    PDUMP 1,X'FFFE'            IF LEVEL = 1 THEN ABORT THE RUN
         EOJ
FINISH   BALR  2,0                  LOAD A USING REGISTER
         USING *,2                  INFORM THE COMPILER OF THIS FACT
F        BC    X'0',PAST            IF CALLED EXTERNALLY WILL OPEN
         OPENR PRINTS               PRINTS IF NECESSARY
PAST     MVC   REP(20),MOD          CHANGE THE HEADING
         MVC   NAME(52),NAME-1      PROPOGGATE BLANKS
         CALL  INFO,(DATA)          GET JOB NAME,DATE+TIME
         MVC   NAMJ(8),DATA         MOVE NAME INTO PRINT LINE
         MVC   DATE(8),DATA+8       MOVE DATE INTO PRINT LINE
         MVC   TIME(8),DATA+16      MOVE TIME INTO PRINT LINE
         CNTRL PRINTS,SK,1
         PUT   PRINTS               WRITE
         L     3,FLAG               CHECK IF ANY PROGRAMMER ERRORS
         BCT   3,NONE               IF NONE, THEN NO DUMP
         PDUMP 1,X'FFFE'            DUMP IF ANY PROG ERRORS(NEG NOS.)
NONE     CLOSER PRINTS              CLOSE THE FILE
         EOJ                        RETURN TO MONITOR
PRINTS   DTFPR                                                            X
               BLKSIZE=132,                                               X
               CONTROL=YES,                                               X
               DEVADDR=SYSLST,                                            X
```

```
                  DEVICE=1403,                                              X
                  IOAREA1=LINE,                                             X
                  MODNAME=THIS,                                             X
                  RECFORM=FIXUNB
PJEC      DS      1D
FLAG      DC      F'2'
PARLST    DS      1F
LINE      DC      8CL1' '
          DC      CL10'***  AIMS '
REP       DC      CL20'ERROR   ***  ROUTINE '
NAME      DC      CL6'      '
          DC      CL6' SEV. '
LEVEL     DC      CL1' '
          DC      CL4' NO.'
NUM       DC      5CL7'       '
          DC      CL5' JOB '
NAMJ      DC      CL9'         '
DATE      DC      CL9'         '
TIME      DC      CL8'        '
          DC      20CL1' '
MOD       DC      CL20'END OF JOB  ***      '
DATA      DC      6F'0'
PATL      DC      XL2'F020'
LEVL      DC      XL2'FFFF'
PATL      DC       XL7'40402020202120'
          END
/*
/*
// EXEC ASSEMBLY
THIS      PRMOD                                                             X
                  SEPASMB=YES,                                              X
                  CONTROL=YES,                                              X
                  RECFORM=FIXUNB
          END
/*
// EXEC ASSEMBLY
INFO      START
          COME ENTER
          PRINT NOGEN
JOB       DS      2F
DATE      DS      2F
TIME      DS      2F
PAR       DS      1F
          DS      3C
PAT       DC      10X'F020202148202C4B2020'
WORK      DS      10C
INTIME    DS      1F
ENTER     L       7,0(0,1)
          GETIME STANDARD
          ST      1,INTIME
          MVC     WORK(10),PAT
          ED      WORK(10),INTIME
          MVC     TIME(8),WORK+2
          COMRG
          MVC     DATE(8),0(1)
          MVC     JOB(8),23(1)
          MVC     0(24,7),JOB
          PRINT GEN
          HOME
```

```
              END
/*
// EXEC ASSEMBLY
LSTCODE   START
          COME  ENTER
CHARS     DC    C'A B C D E F G H I J '
LEGAL     DC    X'0000F300'
WORK      DS    1D
          CNOP  0,4
CHECKS    DC    X'00000002'
          DC    X'00000004'
          DC    X'00000008'
          DC    X'00000010'
          DC    X'00000020'
          DC    X'00000040'
          DC    X'00000080'
          DC    X'00000100'
          DC    X'00000200'
          DC    X'00000400'
ENTER     MVI   WORK,C' '
          MVC   WORK+1(7),WORK
          SR    3,3
          SR    4,4
          L     5,0(0,1)
          LA    6,2(0,0)
          LA    7,18(0,0)
          LA    8,CHECKS
LOOP      LH    9,0(0,5)
          N     9,0(0,8)
          BC    X'8',NOT
          LA    10,CHARS(4)
          A     3,=F'2'
          C     3,=F'8'
          BC    X'2',MANY
          LA    11,WORK-2
          AR    11,3
          MVC   0(2,11),0(10)
          BC    X'F',NOT
MANY      MVI   WORK+7,C'*'
          BC    X'F',D11
NOT       A     8,=F'4'
          BXLE  4,6,LOOP
D11       LH    9,0(0,5)
          N     9,LEGAL
          BC    X'8',DO
          MVI   WORK+7,C'X'
DO        TM    1(5),X'01'
          BC    X'01',DONE
          MVI   WORK+7,C'X'
DONE      L     3,4(0,1)
          MVC   0(8,3),WORK
          HOME
          END
/*
 INCLUDE ILFSSQRT
   PHASE AIMSHEAD,*
// EXEC FFORTRAN
      SUBROUTINE HEADER
C
```

AIMS III SOURCE STATEMENT LISTING

```
C   THE PURPOSE OF THE HEADER FILE IS TO PROVIDE THE SYSTEM WITH A
C DESCRIPTION OF THE COURSE IT IS TO HANDEL. THERE MUST BE ONE HEADER
C FOR EACH DECK ( I.E. TEST ) IN THE COURSE. THIS RECORD CONTAINS THE
C NUMBER OF QUESTIONS IN THE TEST , THE CORRECT ANSWERS,.THE NO OF
C PERMISSABLE RESPONSES TO EACH QUESTION , THE MAXIMUM NO. OF SELECTIONS
C IN THE TEST , ETC.  IN ADDITION, THIS RECORD CONTAINS POINTERS TO THE
C QUESTION AND DIRECTORY FILES . THE PTR TO THE QUESTION FILE ALLOWS
C ACCESS TO A GIVEN QUESTION RECORD MERELY BY ADDING THE QUESTION NO.
C -1 TO THE PRT. THE PTR TO THE DIRECTORY FILE DESCRIBES WHICH TO'S
C ARE WITHIN THIS DECK.
C
C   THIS ROUTINE PROVIDES FOR CREATION, ADDITION, UPDATE, AND LISTING
C OF THE HEADER FILE. LOGICALLY THERE ARE 2 FILES , A SCRATCH AND A
C PERMANENT FILE. RECORDS IN THE SCRATCH FILE ARE NOT USED BY THE SYSTEM
C , THEY ARE MERELY STORED THERE. LOGICALLY   RECORDS 1 - 400 (NLESS *
C NDECK ) BELONG TO THE PERMANET FILE WHILE RECORDS 401 TO 800
C ARE THE TEMPORARY FILE. TOGETHER THEY CONSTITUE THE 800 RECORDS
C OF DATA SET REFERENCE NUMBER 6.
C
C   THERE ARE 3 LOGICAL SECTIONS TO THIS ROUTINE.
C 1)  THE FIRST HANDLES RECORDS IN THE TEMPORARY FILE. ITS FIRST SECTION
C PLACES RECORDS IN THE TEMPORARY FILE IN SORTED ORDER, DELETES OR
C REPLACES RECORDS ((IN TEMP.). IT MAY ALSO UPDATE THE ANSWERS (THIS IS
C THE ONLY MODIFICATION PERMITTED TO HEADERS IN THE PERMANENT FILE) OF
C RECORDS IN THE PERMANENT FILE. THE SECOND PORTION IS A SEQUENCE CHECK
C WHICH INSURES THAT THERE ARE NO MORE THAN 10 TEST / LESSON AND THAT
C LESSONS IN TEMP ASCEND BY 1 AND START WITH THE FIRST LESSON FOLLOWING
C THOSE IN PERMANENT FILE. IT ALSO BUILDS THE MAP OF THE TEMPORARY FILE
C FOR THE SYSTEM FILE.
C
C 2)  THIS SECTION CHECKS THE ERROR FLAG FROM THE SYSTEM FILE. IF IT IS
C OFF, IT TRANSFERS RECORDS TO THE PERMANENT FILE, INITIALIZES THEIR
C QUESTION RECORDS, AND SETS UP THE QUESTION PINTERS.  IT BUILDS THE
C MAP OF THE PERMANENT FILE FOR THE SYSTEM FILE.
C
C 3)  THIS SECTION LISTS BOTH THE PERMANENT AND TEMPORARY FILES.
C
C   THE SYSTEM FILE WILL CONTAIN THE FOLLOWING MAP OF THE HEADER FILE.
C         RECORD 2
C                 WD 1 = HIGHEST LESSON IN PERMANENT
C                 WD 2 = NUMBER OF RECORDS IN PERMANENT
C                 WD 3 = HIGHEST LESSON IN TEMPORARY
C                 WD 4 = NUMBER OF RECORDS IN TEMPORARY
C                 WD 5 = ERROR FLAG (LAST EXECUTION OF SECTION 1 LEFT NO
C                        ERRORS IN TEMPORARY,THEN = 0 , ELSE = 1 )
C
C         RECORD 3
C                 WD I = RECORD & OF 1ST HEADER FOR LESSON I IN THE
C                        PHYSICAL FILE.
C
C         RECORD 4
C                 WD I = NUMBER OF RECORDS IN THE FILE FOR
C                        LESSON I
C
C         RECORD 5
C                 WD 1 = NO. OF RECORDS CURRENTLY IN THE QUESTION FILE
C
      COMMON/SYSTEM/NLESS,NDECK,NREX,NQUEST,NSTUD
      COMMON/FILES/IN1(2),IPRINT,IN2(2),IFILE,IN3,IQUEST,IN4(3),ISYS,
      ATS III SOURCE STATEMENT LISTING

     2   IR5,ITAPE,IR6
      INTEGER CARD
      INTEGER * 2  RECORD(65,100) ,  RCORD2(35) , PTRS(50) , NOS(50)
      INTEGER * 2  LPERM,NPERM,LTEMP, NTEMP, LPRFLG
      INTEGER * 2  SRTMAP(4,400) , DELET(3) , DROP , ZERO , CHECK
     2  , IRECD(4) , QRECNO ,   QSTION(70)
      INTEGER RNAME(2) , EFLAG , RECD , RC2  , SNAME(2)
      DATA RNAME/'READ','ER'/, DELET/'DE','LE','T '/,DROP/1/, ZERO/0/
      DATA   QSTION/40*0,30*'  '/   ,SNAME/'STOR','E'/
      INTEGER * 2 NUMBER(2), LESSON, SEGMNT, TYPE , COURSE , SEQNCE ,
     2   SORT , SELECT , GRADES , GUESTS , RESPNS(43) , LABEL(2),QLAB(2)
     3   ,QPTR
      INTEGER  HEADNG(4,2) , ANSWER(2,40) , BLANK , RED , ISTART(2)  ,
     2  DATE(6) , DROPPD(2,2), COND(2)
      INTEGER * 2 QREX(70)
      DATA   HEADNG/'PERM','ANEN','T FI','LE','TEMP','ORAR','Y FI','LE'/
      DATA  BLANK/'    '/ , ISTART/1,401/,LABEL/'.','S.'/,QLAB/' 5','10'
     2/, DROPPD/' ',' ','DELE','TED' / ,COND/'A-OK','STOP'/
      EQUIVALENCE  (LESSON ,  RCORD2(1) )  ,  (SEGMNT ,  RCORD2(3)  )
     2 ,  ( TYPE ,  RCORD2(4) )  , ( COURSE ,  RCORD2(5) )
     3 ,  (SEQNCE ,  RCORD2(6) ) , (SORT,  RCORD2(10) )
     4 ,  (SELECT ,  RCORD2(11) ) , (GRADES,  RCORD2(12) )
     5 ,  (GUESTS ,  RCORD2(13) ) , ( RESPNS(1) ,  RCORD2(16)  )
     6 ,  (QPTR  ,   RCORD2(8) )
C
C FIRST OBTAIN THE HEADER MAP FORM THE SYSTEM FILE
C
          REWIND ITAPE
          READ(ISYS'2) LPERM, NPERM, LTEMP, NTEMP
          READ(ISYS'3) PTRS
          READ(ISYS'4) NOS
C SET ERROR FLAG TO 0
          EFLAG = 0
          NEW   = 0
          NCHECK = NTEMP
C AND READ A BUNCH OF RECORDS(UP TO 100 AT A TIME )
  200     CALL CTIO(RECORD,NUMB,LC1)
          IF(NUMB.LT.1) GO TO 2100
C
C NOW WORK WITH THEM
C
          DO 2000 CARD = 1,NUMB
C IF THE LESSON IS ALREADY IN PERMANENT STOR IT'S MERELY A CHANGE OF
C ANSWERS
          IF (RECORD(1,CARD).GT.LPERM) GO TO 500
C NOW WE MUST FIND THE RECORD IN PERMANENT,SO LOOK IN THE MAP
          LESS = RECORD(1,CARD)
          I1 = PTRS(LESS)
          I2 = NOS(LESS) + I1 -1
          DO 400 NR = I1 , I2
          READ(IFILE'NR)  RCORD2
          DO 300 J = 3,4
          IF( RCORD2(J).NE.RECORD(J,CARD) ) GO TO 400
  300     CONTINUE
C WE'VE FOUND A MATCH CHECK OTHER PARAMETERS
          IF( RCORD2(5).NE.RECORD(5,CARD)) GO TO 350
          IF( RCORD2(6).NE.RECORD(6,CARD))GO TO 350
          IF( RCORD2(10).NE.RECORD(10,CARD)) GO TO 350
          IF( RCORD2(13).NE.RECORD(13,CARD)) GO TO 350
```

```
C ALL CHECKS SO PERFORM UPDATE
           RCCRD2(11) = RECORD(11,CARD)
           RCCRD2(12) = RECORD(12,CARD)
          J2 = QUESTS + 17
          DO 325 J = 18,J2
          J3 =  J - 18 + QPTR
          READ (IQUEST'J3) QREX
          RCCRD2(J) = RECORD(J,CARD)
          QREX(12) =  RCCRD2(J)
   325    WRITE(IQUEST'J3) QREX
C WRITE THE NEW RECORD, AND GO WORK WITH NEXT ONE
          WRITE(IFILE'NR)  RCCRD2
          GO TO 2000
   350    J =  RCCRD2(3)
          K =  RCCRD2(4)
          CALL ERROR(RNAME,0,1,NR,LESS,J,K)
          GO TO 2000
   400    CONTINUE
          J = RECORD(3,CARD)
          K = RECORD(4,CARD)
          CALL ERROR(RNAME,0,2,LESS,J,K)
          GO TO 2000
C ITS NOT AN UPDATE OF PERMANENT SO IT MUST EFFECT TEMPORARY
   500    CONTINUE
C SEE IF ITS A DELET RECORD
          DO 550 I =  1,3
          IF(RECORD(I+6,CARD).NE.DELET(I)) GO TO 1000
   550    CONTINUE
C ITS A DELET SO FIND THE RECORD
          LESS = RECORD(1,CARD)
          I1 = PTRS(LESS)
          I2 = NOS(LESS) + I1 - 1
          IF(I1.LT.401.OR.I2.LT.I1) GO TO 900
   600    DO 800 NR = I1 , I2
          READ(IFILE'NR)  RCCRD2
          DO 700  I = 3,4
          IF( RCCRD2(I).NE. RECORD(I,CARD) ) GO TO 800
   700    CONTINUE
C WE'VE FOUND THE MATCH SO SET THE DELETE FLAG, AND GO TO NEXT RECORD
          DROP = 1
          WRITE(IFILE'NR)  RCCRD2,DROP
          NCHECK = NCHECK - 1
          GO TO 2000
   800    CONTINUE
C WE HAVEN'T FOUND THE RECORD
   900    I = RECORD(3,CARD)
          J = RECORD(4,CARD)
          EFLAG = 1
          CALL ERROR(RNAME,0,3,LESS,J,K)
          GO TO 2000
C THE RECORD ISN'T A DELET, SO ITS A REPLACE OR ADD
  1000    CONTINUE
C  CHECK IF THERE ARE ANY QUESTIONS
          IF (RECORD(13,CARD).GE.1 ) GO TO 1100
          LESS = RECORD(1,CARD)
          I= RECORD(3,CARD)
          J= RECORD(4,CARD)
          EFLAG = 1
          CALL ERROR(RNAME,0,4,LESS,I,J)
```

FILMED FROM BEST AVAILABLE COPY

```
          GO TO 2000
 1100     LESS = RECORD(1, CARD)
          I1 = PTRS(LESS)
          I2 = NOS(LESS) + I1 -1
C IF NONE FOR THIS LESSON ITS AN ADD
          IF(I1.LT.401.OR.I2.LT.I1) GO TO 1500
C OTHERWISE SEE IF ITS A REPLACE
          DO 1200 NR = I1 , I2
          READ(IFILE'NR)  RCORD2
          DO 1150 I = 3,4
          IF ( RCORD2(I) .NE. RECORD(I,CARD) ) GO TO 1200
 1150     CONTINUE
C YES ITS A REPLACE
          WRITE(IFILE'NR) (RECORD(I,CARD),I=1,65),ZERO
          GO TO 2000
 1200     CONTINUE
C NO THIS IS AN ADDITION
 1500     CONTINUE
C  SEE IF THERE'S MORE ROOM IN THE FILE
          I = NEW + NCHECK + NPERM
          IF(I. LT. 400 ) GO TO 1600
          I=RECORD(3,CARD)
          J=RECORD(4,CARD)
          EFLAG = 1
          CALL ERROR(RNAME,0,5,LESS,I,J)
          GO TO 2000
 1600     CONTINUE
          NEW = NEW + 1
          SRTMAP(1,NEW) = RECORD(1,CARD)
          SRTMAP(2,NEW) = RECORD(3,CARD)
          SRTMAP(3,NEW) = RECORD(4,CARD)
          SRTMAP(4,NEW) = 0
          WRITE(ITAPE) (RECORD(I,CARD),I = 1,65)
 2000     CONTINUE
          IF (LCI .EQ . 0) GO TO 200
C IF LAST CARD READ,WRITE OLD
C TEMPORARY RECORDS TO SORT TAPE
 2100     IF(NTEMP.LE.0) GO TO 2300
          I1 = 401
          I2 = NTEMP + I1 -1
          DO 2200 I = I1 , I2
          READ(IFILE'I)  RCORD2,CHECK
          IF( CHECK.NE. 0) GO TO 2200
          NEW = NEW + 1
           SRTMAP(1,NEW) =  RCORD2(1)
          SRTMAP(2,NEW) =  RCORD2(3)
          SRTMAP(3,NEW) =  RCORD2(4)
          SRTMAP(4,NEW) = 0
          WRITE(ITAPE) RCORD2,CHECK
 2200     CONTINUE
 2300     IF(NEW.LE.0) GO TO 4200
C BEGIN TO SORT THE RECORDS, IF ANY
          END FILE ITAPE
          REWIND ITAPE
          DO 3000 IPOS = 1, NEW
          DO 2400 I = 1, NEW
          IF(SRTMAP(4,I  ) .NE. 0) GO TO 2400
          IWH = I
          GO TO 2500
 2450    CONTINUE
         GO TO 3100
 2500    CONTINUE
         IF(IWH.NE. NEW) GO TO 2600
         SRTMAP(4,IWH) = IPOS
         GO TO 3100
 2600    CONTINUE
         J = IWH + 1
         DO 2900 I = J,NEW
         IF(SRTMAP(4,I).NE.0) GO TO 2900
         IF(SRTMAP(1,IWH) - SRTMAP(1,I) ) 2900 , 2700 , 2850
 2700    IF(SRTMAP(2,IWH) - SRTMAP(2,I) ) 2900 , 2750 , 2850
 2750    IF(SRTMAP(3,IWH) - SRTMAP(3,I) ) 2900 , 2900 , 2850
 2850    CONTINUE
         IWH = I
 2900    CONTINUE
         SRTMAP(4,IWH) = IPOS
 3000    CONTINUE
 3100    CONTINUE
C MAKE SURE TAPE IS REWOUND
         REWIND ITAPE
C AND READ RECORDS BACK ON TO THE DISK
         DO 3200 I = 1  , NEW
         J = SRTMAP(4,I) + 400
         READ(ITAPE)  RCORD2
 3200    WRITE(IFILE'J)  RCORD2 , ZERO
C
C WE'VE FINISHED BUILDING AND UPDATING, SO NOW BEGIN SEQUENCE CHECK
C INITIALIZE
C
 3300    LAST = LPERM
         KOUNT = 0
         I1 = LAST + 1
         DO 3350 I = I1 , NLESS
         PTRS(I) = 0
 3350    NOS(I) = 0
         I2 = 401 + NEW - 1
         DO 4000 I = 401, I2
         READ(IFILE'I) IRECD
         IF(KOUNT.EQ.0) GO TO 3300
         IF( IRECD(1) - LAST ) 3400, 3500, 3700
 3400    CONTINUE
         J = IRECD(1)
         CALL ERROR(RNAME,0,-1,I,LAST,J)
         GO TO 3950
 3500    KOUNT = KOUNT + 1
         IF(IRECD(3).NE.LAST3) GO TO 3600
         IF(IRECD(4).NE.LAST4) GO TO 3600
         EFLAG = 1
         J = I - 400
         CALL ERROR(RNAME,0,6,J,LAST,LAST3,LAST4)
 3600    CONTINUE
         IF(KOUNT.LE.NDECK) GO TO 3950
         EFLAG = 1
         CALL ERROR(RNAME,0,7,LAST,LAST3,LAST4,KOUNT)
         GO TO 3950
 3700    CONTINUE
C STORE THE NUMBER OF RECDS IN THE LAST LESSON
         NOS(LAST) = KOUNT
```

```
        NIS III SOURCE STATEMENT LISTING

 C CHECK THE LESSON NUMBER
  3800    CONTINUE
          LAST = LAST + 1
          IF( LAST.EQ.IRECD(1)) GO TO 3900
          J = IRECD(1)
          K = I - 400
          EFLAG = 1
          CALL ERROR(KNAME,0,4,K,LAST,J)
          LAST = IRECD(1)
  3900    CONTINUE
          PTRS(LAST) = 1
          KOUNT = 1
  3950    LAST3 = IRECD(3)
          LAST4 = IRECD(4)
  4000    CONTINUE
 C STORE KOUNT FOR THE LAST DECK
          NCS(LAST) = KOUNT
          LTEMP = LAST
          NTEMP = NEW
          GO TO 4400
  4200    CONTINUE
          LTEMP = 0
          NTEMP = 0
          IF(LPERM.EQ.NLESS) GO TO 4400
          IL = LPERM + 1
          DO 4300 I = IL , NLESS
          NCS(I) = 0
  4300    PTRS(I) = 0
  4400    CONTINUE
          ERRFLG = EFLAG
          WRITE(ISYS'2) LPERM, NPERM, LTEMP, NTEMP , ERRFLG
          WRITE(ISYS'3) PTRS
          WRITE(ISYS'4) NCS
  4500    CONTINUE
          GO TO 6800
          ENTRY STORE
 C THIS SECTION MUST CHECK THE ERROR FLAG ON THE DISK
 C IF ON NO ACTION WILL BE TAKEN, IF OFF ALL RECORDS WILL BE MOVED FROM
 C TEMPORARY TO PERMANENT
 C
          READ(ISYS'2) LPERM, NPERM, LTEMP, NTEMP, ERRFLG
          IF(ERRFLG .EQ. 0) GO TO 5000
          CALL ERROR(SNAME,0,1)
          GO TO 6800
 C READ HEADER MAP
  5000    IF(NTEMP.LE.0)GO TO 6800
          READ(ISYS'3) PTRS
          READ(ISYS'4) NCS
 C ALSO READ NUMBER OF QUESTION RECORDS
          READ(ISYS'5) QRECNO
          KOUNT = 0
          LAST = LPERM
          RC2 = 401 + NTEMP - 1
          DO 6000 RECD = 401, RC2
 C INCREMENT RECORD NUMBER
          NPERM = NPERM + 1
          READ(IFILE'RECD)  RCORD2
 C IF FIRST NONE TO STORE IN NCS    1
          IF(KOUNT.EQ.0 ) GO TO 5100
```

```
C CHECK FOR LESSON CHANGE
            IF(LAST.EQ. RCORD2(1)) GO TO 5300
            NOS(LAST) = KOUNT
  5100      LAST =  RCORD2(1)
            PTRS(LAST) = NPERM
            KOUNT = 1
            GO TO 5500
C THIS IS THE SAME LESSON
  5300      CONTINUE
            KOUNT = KOUNT + 1
C NOW WE PLACE THE RECORD IN PERM, BUT FIRST SET UP QUESTION RECORDS
  5500      CONTINUE
            IQ2 =  RCORD2(15)
            QSTION(1) =  RCORD2(1)
            QSTION(2) =  RCORD2(3)
            QSTION(3) =  RCORD2(4)
            QSTION(4) = RCORD2(5)
            DO 5800 I = 1 , IQ2
            I1 = QRECNO + I
            QSTION(5) = 1
            QSTION(12) =  RCORD2(I+17)
            IF( I1. LE. 4000) GO TO 5800
C NO MORE ROOM
            J = NPERM
            K =  RCORD2(1)
            L =  RCORD2(3)
            M =  RCORD2(4)
            CALL ERROR(SNAME,1,-1,J,K,L,M)
  5800      WRITE(IQUEST'I1) QSTION
C SET UP QUESTION PTR
             RCORD2(7) = 0
             RCORD2(8) = QRECNO + 1
            QRECNO = QRECNO + IQ2
            WRITE(IFILE'NPERM)  RCORD2
            DROP = 1
            WRITE(IFILE'RECD)  RCORD2 , DROP
  6000      CONTINUE
C COMPLETE MAP FOR LAST HEADER
            NOS(LAST) = KOUNT
C REWRITE MAP, AND FALL THROUGH TO LIST
            LTEMP = 0
            NTEMP = 0
            LPERM = LAST
            ERRFLG = 1
            WRITE(ISYS'2) LPERM, NPERM, LTEMP, NTEMP, ERRFLG
            WRITE(ISYS'3) PTRS
            WRITE(ISYS'4) NOS
            WRITE(ISYS'5) QRECNO
            ENTRY HOLIST
  6800      CONTINUE
            READ(ISYS'2) LPERM, NUMBER(1), LTEMP, NUMBER(2), ERRFLG
            IPAGE = 0
            CALL INFO(DATE)
            DO 9000 RED = 1 , 2
            IH = 0
            IF(NUMBER(RED).LE.0) GO TO  9000
            I1 = ISTART(RED)
            I2 = I1 - 1  + NUMBER(RED)
            DO 6000 IRX = I1, I2
```

```
      WI·S III SOURCE STATEMENT LISTING

         IF( IH .GT. 0) GO TO 7000
         IPAGE = IPAGE + 1
         WRITE(IPRINT,6999) DATE ,(HEADNG(J,RED),J=1,4),IPAGE
         IH = 4
  7000   CONTINUE
         IH = IH - 1
C READ THE HEADER RECORD
         READ(IFILE'IRX)  RCORD2 , DROP
         DO 7500 I = 1 , QUESTS
  7500   CALL LSTCDE(RESPNS(I),ANSWER(1,I))
         J2 = IRX - I1 + 1
         J3 = SCRG + 1
         J4 = DROP + 1
         WRITE (IPRINT,7999) COURSE , LESSON , SEGMNT , TYPE ,
     2  SEQNCE , LABEL(SEQNCE) , J2 , QUESTS  , SLAB(J3) , GRADES
       , SELECT , (DROPPD(J,J4),J=1,2)
         IF (QUESTS.GT. 12) GO TO 7700
         WRITE(IPRINT,8001) (J,J=1,QUESTS)
         WRITE (IPRINT,8002) ((ANSWER(J,K),J=1,2),K=1, QUESTS)
         WRITE (IPRINT,8003)
         WRITE (IPRINT,8003)
         WRITE (IPRINT,8003)
         WRITE(IPRINT,8003)
         GO TO 8000
  7700   IF (QUESTS.GT.24) GO TO 7800
        WRITE (IPRINT,8001) (J,J=1,12)
         WRITE (IPRINT,8002) ((ANSWER (J,K),J=1,2),K=1,12)
         WRITE (IPRINT,8001) (J,J=13,QUESTS)
         WRITE(IPRINT,8002) ((ANSWER(J,K),J=1,2),K=13, QUESTS)
         WRITE (IPRINT,8003)
         WRITE (IPRINT,8003)
         WRITE(IPRINT,8003)
         GO TO 8000
  7800   IF (QUESTS.GT.36) GO TO 7900
         WRITE (IPRINT,8001) (J,J=1,12)
         WRITE (IPRINT,8002) ((ANSWER (J,K),J=1,2),K=1,12)
         WRITE (IPRINT,8001) (J,J=13,24)
         WRITE (IPRINT,8002) ((ANSWER(J,K),J=1,2),K=13,24)
         WRITE (IPRINT,8001) (J,J=25,QUESTS)
         WRITE (IPRINT,8002) ((ANSWER (J,K),J=1,2),K=25,QUESTS)
         WRITE (IPRINT,8003)
         WRITE(IPRINT,8003)
         GO TO 8000
  7900   CONTINUE
         WRITE (IPRINT,8001) (J,J=1,12)
         WRITE (IPRINT,8002) ((ANSWER(J,K),J=1,2),K=1,12)
         WRITE (IPRINT,8001) (J,J=13,24)
         WRITE (IPRINT,8002) ((ANSWER(J,K),J=1,2),K=13,24)
         WRITE (IPRINT,8001) (J,J=25,36)
         WRITE (IPRINT,8002) ((ANSWER(J,K),J=1,2),K=25,36)
         WRITE (IPRINT,8001) (J,J=37,QUESTS)
         WRITE (IPRINT,8002) ((ANSWER(J,K),J=1,2),K=37,QUESTS)
         WRITE(IPRINT,8003)
  8000   CONTINUE
         IF( IH . GT. 0)  GO TO 8500
         IPAGE = IPAGE + 1
         WRITE (IPRINT,6999) DATE , (HEADNG(J,RED),J=1, 4),IPAGE
  8500   CONTINUE
         WRITE (IPRINT,8501) NUMBER(RED)
         IF( RED.EQ. 1) GO TO 9000
         K = ERRFLG + 1
         WRITE ( IPRINT,8999) COND(K)
 9000    CONTINUE
         RETURN
 6599    FORMAT(1H1  ,  7X , '*** ','AIMS COURSE DESRIPTION' ,
     2   '  ***' ,    50X   , 'JOB' , 2X , 2A4  ,  2X ,2A4  ,
     3    1X  ,  2A4     /   13X ,  4A4   , 61X   , 'PAGE' ,
     4    I4  /  / )
 7599    FORMAT( 10X , 'COURSE' , 1X ,  I2 ,   11X  , 'LESSON', 1X  ,
     2   I2 ,  11X ,  'SEGMENT', 1X   ,  I2  , 10X  , 'TYPE' ,  1X  ,
     3   I1 ,   4X ,  'CONSISTS OF' ,  I2  ,  1X  ,  'CARD'   ,  A2 ,
     4   5X ,  'RECORD' , I4  /
     5   10X,  'THERE ARE',  I3 , 1X , 'QUESTIONS, EACH OF UP TO'   ,
     6   1X,  A2 , 1X  ,'ANSWERS.', I4 , 1X , 'WILL BE GRADED.'  ,
     7   2X , 'THERE ARE' ,  I4   ,  1X ,'POSSIBLE SELECTIONS.'  , 2X ,
     8    2A4    )
 5501    FORMAT( 11(/)  ,  13X    , 'THERE ARE' , I4   ,  1X,
     2   'RECORDS IN THIS FILE'   )
 8001    FORMAT(3X,11(I2,9X),I2)
 8002    FORMAT(3X,11(2A4,3X),2A4)
 8003    FORMAT(/)
 8999    FORMAT( 13X , 'THE CONDITION IS' , 1X , A4)
         END
         SUBROUTINE CTIC(RECD,NUM,LCI)
         COMMON/FILES/IW1(12),IR , IW2(2)
         INTEGER * 2 RECD(65,100)
       REWIND IR
         LCI = 1
         NUM = 0
          DO 500 I = 1,100
        READ(IR,END=1000) (RECD(L,I),L=1,65)
         NUM = I
  500    CONTINUE
         LCI = 0
 1000    CONTINUE
       REWIND IR
          RETURN
         END
/*
 PHASE AIMSCLAS,AIMSHEAD
// EXEC FFORTRAN
       SUBROUTINE INPUT(INDEV)
C
C
C  THERE ARE 3 SECTIONS OR ENTRY POINTS TO THIS ROUTINE,EACH WITH ITS
C    OWN FUNCTION. THE MAIN PURPOSE OF THIS ROUTINE IS HANDLING OF THE
C    STUDENT BACKGROUND FILE. THIS FILE SERVES AS A MASTER FILE FOR THE
C    STUDENTS, CONTROLLING EACH PROCESS RUN, AND CONTAINING ALL INFORMA-
C    TION PERTAINING TO EACH STUDENT(OTHER THAN HIS GRADES) . THE
C    STUDENT NUMBER(BY WHICH THE SYSTEM REFERS TO A PARTICULAR STUDENT)
C    IS THE RECORD NO. IN THIS FILE. THE 1ST WORD OF THE FIRST RECORD
C    IN THE SYSTEM FILE CONTAIN THE NO. OF RECORDS IN THIS FILE, THE 2ND
C    WORD IS THE N OF STUDENTS(NOT COUNTING DROPS).
C  THE 3 FUNCTIONS OF THIS ROUTINE ARE
C    1) CREATING OR ADDING TO THE FILE
C        IN THIS FUNCTION THE VALUES READ FOR A STUDENT ARE USED TO
C        CALCULATE A CAPABILITY INDEX FOR EACH STUDENT + PLACE IT IN HIS
C        RECORD
```

```
C     2) DROPPING STUDENTS
C          SETTING THE DROP FLAG IN A STUDENT'S RECORD TO 1
C     3) LISTING THE FILE
C          ALL STUDENTS INCLUDING THOSE DROPPED WILL BE LISTED. THERE
C       ARE CURRENTLY 2 VERSIONS OF THIS ROUTINE ONE FOR EACH USER
C     THEY DIFFER IN THE ALGORITHM FOR CALCULATING THE CAPABILITY
C     INDEX, AND THE FORMATS FOR LISTING THE FILE
C
C   THE FOLLOWING COMMONS SET UP THE DATA SET REFERENCE NOS., AND THE
C   SYSTEM PARAMETERS, THE INPUT UNIT IS THE PARAMETER
        COMMON/SYSTEM/NLESS,NDECK,NREA,NQUEST,NSTUD
        COMMON/FILES/IW1(2),IPRINT,IW2(5),IFILE,IW3(2),ISYS,IW4(3)
C   THE NO. OR RECDS PRESENTLY IN THE FILE, THE NO. OF STUDENTS(FROM THE
C    SYSTEM FILE
        INTEGER * 2 NINFIL,NINCRS
C
C   THE FOLLOWING INPUT VALUES ARE EQUIVALENCED TO THE CORRECT POSITIONS
C    IN THE OUTPUT RECORD
C
        INTEGER * 2 NAME(13),IDNO(5),SATM,SATV,AVR,RANK,ALGE,GEOM,TRIG,
       1 ALGI,CALC,PHYS,IQ,READ,COMENT(6), COURSE , CAPIN , RECORD(65)
       2  , CHEM ,NAVRNK , MTHACH , STUDNO  , SECOND(65)
        EQUIVALENCE (COURSE,RECORD(5)) , (NAME(1),RECORD(6))     ,
       1 (IDNO(1),RECORD(19)) , (CAPIN,RECORD(25)) , (SATM,RECORD(29))    ,
       2 (SATV,RECORD(30)) , (AVR,RECORD(31)) , (RANK,RECORD(32))        ,
       3 (ALGE,RECORD(33)) , (GEOM,RECORD(34)) , (TRIG,RECORD(35))       ,
       4 (ALGI,RECORD(36)) , (CALC,RECORD(37)) , (PHYS,RECORD(38))       ,
       5 (IQ,RECORD(39)) , (READ,RECORD(40)) ,  (NAVRNK,RECORD(41))      ,
       6 (COMENT(1),RECORD(42)) , (MTHACH ,RECORD(39) ) ,(CHEM,RECORD(38))
       7     , (STUDNO, RECORD(2) )
        DATA RECORD/41*0 , 14 * ' ' ,  10 * 0 /
        DIMENSION NERR(2)
        DATA NERR/'INPU','T'/
C
C   THESE WORDS ARE FOR THE DROP SECTION
C
        INTEGER DNAM(2)
        DATA DNAM/'DROP',' '/
        INTEGER * 2 DATE(12) , STUDNM(13),DROPFG
        EQUIVALENCE (DROPFG,RECORD(24))
        INTEGER * 2 BLANK
        DATA BLANK/' '/
C
C   LIST SECTION
C
        INTEGER  * 2 DATIN(4),DATOUT(4)
        EQUIVALENCE (DATIN(1) , RECORD(48)) , ( DATOUT(1), RECORD(52) )
        INTEGER DROPD(2)
        DATA DROPD/'  ','YES'/
C   CLEAR RECORD AREA
C   THIS IS THE MAIN ENTRY FOR FILE CREATION
C
C   OBTAIN THE NOS. OF RECORDS AND STUDENTS AND THE DATE
        NOS = 0
        READ(ISYS'1) NINFIL,NINCRS
        CALL INFO(DATE)
C   AND BEGIN TO INPUT STUDENTS
    200 CONTINUE
        DO 10 I = 1,41
        AIMS III SOURCE STATEMENT LISTING

   10 RECORD(I) = 0
      DO 20 I = 42,55
   20 RECORD(I) = BLANK
      DO 30 I = 56,65
   30 RECORD(I) = 0
       READ(INDEV,201,END=2800) NAME,IDAC,SATM,SATV,AVR,RANK,ALGE,GEOM,
     1 TRIG,ALGI,CALC,PHYS,IQ,KLAS,COMENT,CCURSE,STUDNO
  201 FORMAT(12A2,A1,  4A2,A1, I3 , I3 ,  I2 , I1 , I2 , I2 , I2 , I2 ,
     1 I2 , I2  ,  I3 ,  I2 , 3X  ,  6A2  ,  I2  ,  I3  )
       NOS = NOS + 1
C
C IF STUDNO ISN'T 0 THIS IS AN UPDATE CARD AND REPLACES AN EXISTANT
C   RECORD
C
       IF (STUDNO .EQ. 0) GO TO 299
C THIS IS AN UPDATE SO CHECK THE NUMBER
       IF (STUDNO.GE.1.AND.STUDNO.LE.NINFIL) GO TO 275
       I=STUDNO
       CALL ERROR(NERR,0,1,NOS,I)
       GO TO 200
C THE NO. IS OK SO READ THE OLD RECORD(DATES,DROPPS,ETC. MUST BE SAVED)
  275 READ(IFILE'STUDNO) SECOND
C CHECK THAT THE NAMES MATCH (ONE STUDENT MAY NOT REPLACE ANOTHER)
       DO 280 I = 1,13
       IF (NAME(I).EQ.SECOND(I+5)) GO TO 280
       J=STUDNO
       CALL ERROR(NERR,0,2,NOS,J,I)
       GO TO 200
  280 CONTINUE
C ALL CHECKS SO WE CAN GO CALCULATE THE CAPABILITY INDEX
       GO TO 400
  299 IF(NSTUD - NINFIL) 300, 300 ,400
C  THERE'S NO MORE ROOM, SO LEAVE
  300 CALL ERROR(NERR,0,3,NOS,NSTUD)
       GO TO 2800
C
C  MORE ROOM ,SO PLACE STUDENT NO IN RECORD CALCULATE CAPIN, AND
C    WRITE OUT THE RECORD
C
  400 CONTINUE
C
C          CAPABILITY INDEX CALCULATIONS DEFFERS FOR EACH USER
C  AS NO INPUT VALUE COULD REALISTICALLY BE 0 , 0 INDICATES
C  MISSING DATA
C
C  THIS ONE IS FOR HEW
C
       C1 = 0.
       C2 = 0.
       SUM1=0.
       SUM2=0.
       DO 600 I = 33,37
       IF( RECORD(I). EQ . 0 ) GO TO 600
       SUM1 = SUM1 + RECORD(I)
       C1 = C1 + 1.
  600 CONTINUE
       IF(SATM.EQ.0) GO TO 1000
  800 CONTINUE
       C2 = C2 + 1.
      SUM2 = SUM2 + (SATM - 200.)/ 6.
 1000 CONTINUE
      IF(SATV.EQ.0) GO TO 1200
      C2 = C2 + 1.
      SUM2 = SUM2 + (SATV - 200.)/ 6.
 1200 CONTINUE
      IF(AVR.EQ.0) GO TO 1400
      C2 = C2 + 1.
      SUM2 = SUM2 + AVR
 1400 CONTINUE
      IF(RANK.EQ.0) GO TO 1600
      C2 = C2 + 1.
      SUM2 = SUM2 + (RANK +1) * 10
 1600 CONTINUE
      IF(PHYS.EQ.0) GO TO 1800
      C2 = C2 + 1.
      SUM2 = SUM2 + PHYS
 1800 CONTINUE
      IF(IQ.EQ.0) GO TO 2000
      C2 = C2 + 1.
      SUM2 = SUM2 + IQ /2.
 2000 CONTINUE
      IF(READ.EQ.0) GO TO 2200
      C2 = C2 + 1.
      SUM2 = SUM2 + READ * 7.
 2200 CONTINUE
      IF(C1.EQ.0) GO TO 2400
      C2 = C2 + 1.
      SUM2 = SUM2 +(SUM1/C1)
 2400 CONTINUE
      IF ( C2 .GT.0.9 ) GO TO 2500
      CAPIN = -999
      GO TO 2550
 2500 CAPIN = SUM2 / C2
 2550 CONTINUE
      IF (STUDNO.GT.0) GO TO 2700
      NINCRS = NINCRS + 1
      NINFIL = NINFIL + 1
      STUDNO = NINFIL
      DO 2600 I = 1,4
 2600 DATIN(I) = DATE(I+4)
      WRITE(IFILE'NINFIL) RECORD
      GO TO 200
C THIS WAS AN UPDATE
 2700 CONTINUE
        RECORD(24) = SECOND(24)
        DO 2710 I =26,28
 2710   RECORD(I) = SECOND(I)
        DO 2720 I =48,65
 2720   RECORD(I) = SECOND(I)
C HAVING PRESERVED INFORMATION, REWRITE THE RECORD
        WRITE(IFILE'STUDNO) RECORD
        GO TO 200
C
C  HAVING FINISHED, UPDATE THE SYSTEM FILE AND GO LIST
C
 2800 CONTINUE
      WRITE(ISYS'1) NINFIL,NINCRS
      GO TO 5000
```

AIMS III SOURCE STATEMENT LISTING

```
C
C  THIS SECTION DELETES STUDENTS FROM THE SYSTEM
C
      ENTRY DROP(INDEV)
 3000 CONTINUE
C  GET SYSTEM COUNTS (NO. OF RECORDS,NO. OF STUDENTS)
      READ(ISYS'1) NINFIL,NINCRS
C  GET CURRENT DATE
      CALL INFO(DATE)
      NOS = 0
C
C  BEGIN READING STUDENTS
C
 3300 READ(INDEV,3301,END=4800) STUDNM,STUDNO
 3301 FORMAT(12A2,A1,  52X  , I3 )
      NOS = NOS +1
C  CHECK STUDNO FOR VALIDITY
      IF ( STUDNO.GT.0 .AND. STUDNO.LE. NINFIL ) GO TO 3600
      J = NINCRS
      I=STUDNO
      CALL ERROR(DNAM,0,1,NOS,I,J)
      GO TO 3300
C  READ THE STUDENT RECORD
 3600 READ(IFILE'STUDNO) RECORD
      IF(DROPFG.EQ.0 ) GO TO 4000
      I = STUDNO
      CALL ERROR(DNAM,0,2,NOS,I)
      GO TO 3300
C  IF STILL IN COURSE CHECK THE NAME
 4000 CONTINUE
      DO 4300 I = 1,13
      IF(STUDNM(I) .EQ. NAME(I) ) GO TO 4300
      J = STUDNO
      CALL ERROR(DNAM,0,3,NOS,J,I)
      GO TO 3300
 4300 CONTINUE
C  ALL CHECKS,DROP STUDENT
      DROPFG = 1
C  PLACE DATE DROPPED IN RECORD
      DO 4600 I = 1,4
      DATOUT(I) = DATE(I+4)
 4600 CONTINUE
      WRITE(IFILE'STUDNO) RECORD
C
C  CORRECT COUNT OF STUDENTS
C
      NINCRS = NINCRS - 1
      GO TO 3300
C  WE'VE DROPPED ALL STUDENTS
 4800 WRITE(ISYS'1) NINFIL,NINCRS
C  AND FALL THROUGH TO LIST ROUTINE
C
C  THE PURPOSE OF THIS SECTION IS TO PRODUCE A COURSE LISTING
      ENTRY LIST
 5000 CONTINUE
C  GET THE NO. OF RECORDS, AND OF STUDENTS, AND DATE,ETC.
      READ (ISYS'1) NINFIL,NINCRS
      CALL INFO(DATE)
      III = 0
```

```
       AIMS III SOURCE STATEMENT LISTING

       IPAGE = 0
       DO 9000 I = 1 , NINFIL
       IF( IH .GT. 0 ) GO TO 8000
       IPAGE = IPAGE + 1
       WRITE(IPRINT,7999) DATE,IPAGE
       WRITE(IPRINT,7998)
       IH = 40
  7999 FORMAT(1H1      ,     7X   , '***'   , 2X , 'AIMS COURSE ROSTER' ,  2X,
      1'***',55X,'JOB', 2X,4A2    ,    2X   ,     4A2   ,   1X    ,   4A2        ,
      2   / 91X,    'PAGE' , I4  /  /    /        )
  7998 FORMAT(
      3 1X,  'NAME' ,  22X , 'S.S. NUM.' ,   1X , 'NO.' , 1X ,  'CAP.'
      4   , 1X ,   'SAT' , 2X , 'SAT' ,   1X , 'AVR' ,   4X , 'ALG' , 2X  ,
      5 'ALG' ,   1X , 'GEOM' , 4X , 'CALC' , 5X , 'IQ' ,4X,'COMMENTS'   ,
      6  5X,     'CRSE',1X   , 'DROP' , 1X  , 'DATE' , 5X , 'DATE'      /
      7 41X , 'INDX' ,   1X   ,'MATH' , 1X , 'VERB' , 5X , 'RNK'        ,
      8  1X , 'ELEM' ,   1X   , 'INTR'   , 4X ,'TRIG' ,   4X   ,'PHYS'
      9  3X ,'READ' ,  23X   , 'ENTERED' ,   2X  , 'DROPPED'  /  /    )
  8000 READ(IFILE'I) RECORD
       IH = IH -1
       K = DROPFG + 1
       WRITE(IPRINT,8001) NAME , ICNO, STUDNO , CAPIN,SATM,SATV,AVR,RANK,
      1 ALGE, ALG1, GEOM , TRIG , CALC , PHYS , IQ , READ,
      2 COMENT , COURSE ,DROPD(K) , DATIN ,DATOUT
  8001 FORMAT( 1X , 12A2,A1 ,  1X ,   4A2,A1 ,    1X , I5  ,    2X ,   I2 ,
      2  2X , I5   ,   2X ,  I3 ,   2X , I2  ,   1X , I1   ,  3X , I2  ,
      3  3X ,  I2   ,    2X , I2   , 2X  I2  ,    2X ,  I2   ,  2X , I2  ,
      4  1X ,  I3 ,   1X ,  I2 ,    1X , 6A2    ,  2X  ,   I2   ,  2X,  A3 ,
      5  1X ,  4A2   ,   1X   , 4A2     )
  9000 CONTINUE
       IPAGE = IPAGE+1
       WRITE(IPRINT,7999) DATE,IPAGE
       WRITE(IPRINT,9001) NINFIL,NINCRS
  9001 FORMAT(1X, 9(/)                          ,
      1  13 X    , 'THERE ARE' ,I4   ,' RECORDS IN THE STUDENT FILE'     /
      2  14 X    , 'OF THESE'  ,I4   ,' STUDENTS REMAIN IN THE COURSE '   )
       RETURN
       END
 /*
  PHASE AIMSQUST,AIMSHEAD
 // EXEC FFORTRAN
             SUBROUTINE MBC1(INUNIT,LEVEL)
 C
 C     THE PURPOSE OF THIS ROUTINE IS TO ADD MBC INFORMATION TO THE
 C QUESTION FILE AND THEN CHAIN THE INFORMATION IN SUCH A WAY THAT
 C ALL QUESTIONS RELATED TO THE SAME TO  THIS ROUTINE READS THE SYSTEM
 C FILE ONLY TO ACCESS THE HEADER FILE, IT READS THE HEADERS FROM THE
 C PERMANENT FILE, IN ORDER TO BUILD A MAP OF THE QUESTION FILE IN CORE.
 C THIS MAP ALLOWS A DIRECT ACCESS OF ANY QUESTION RECORD. THUS MBC CARDS
 C MAY BE IN ANY ORDER, THOUGH TIME WILL BE SAVED IF THEY ARE IN LESSON,
 C SEGMENT + TYPE ORDER.
 C
 C     THE FIRST SECTION OF THIS ROUTINE BUILDS THE TABLE, READS THE MBC
 C CARD, READS THE CORRECT QUESTION RECORD , ADDS THE MBC INFORMATION,
 C AND REWRITES THE RECORD.  THIS PREVENTS DESTRUCTION OF ANY QUEST.
 C VALIDITY INFO WHICH MAY BE IN THE RECORD FROM A PROCESS RUN.
 C
 C     THE SECOND SECTION DETERMINES WHERE TO START CHAINING FROM, AND
 C THEN CHAINS FROM THAT POINT.  THIS PREVENTS UN-NECESSARY RECHAINING,
C YET ALLOWS THE SAME ROUTINE TO BE USED FOR INITIAL CHAINING, AND
C SUBSEQUENT UPDATE.
C
C     THE FILES USED BY THIS ROUTINE ARE AS FOLLOWS
C
C     SYSTEM FILE
C 1) RECORDS 2,3, +4 ARE READ IN ORDER TO FACILITATE READING OF THE
C HEADER FILE. RECORD 5 WILL BE READ TO DETERMINE HOW MANY RECORDS
C ARE IN THE QUESTION FILE.
C RECORD 5 WILL CONTAIN THE NUMBER OF DIRECTORY RECORDS CURRENTLY USED.
C     HEADER FILE
C 1) THE HEADER FILE POINTERS ARE READ INTO CORE IN ORDER TO BUILD A
C TABLE WHICH ALLOWS DIRECT ACCESS OF THE QUESTION FILE
C 2) THE DIRECTORY POINTERS ARE CHECKED TO DETERMINE WHERE WE LAST CHAI.
C -ED, AND AS THE DIRECTORY RECORDS ARE CREATED, THEIR NUMBERS ARE
C PLACED IN THE DIRECTORY POINTERS OF THE CORRECT HEADER RECORDS.
C
C
C     QUESTION FILE
C 1) THE QUESTION RECORDS FOR WHICH MBO INFORMATION IS READ ARE
C READ, THE INFO. ADDED, AND ARE THEN REWRITTEN. THUS QUESTION RECORDS
C ARE CREATED IN SUBROUTINE HEADER. THIS ROUTINE ADDS MBO INFORMATION.
C THE PROCESS ROUTINES WILL ADD THE QUESTION VALIDITY INFORMATION
C 2) A PASS WILL BE MADE THROUGH THE QUESTION FILE TO READ THE MBO INFO,
C AND PLACE IT IN A TABLE. CHAINING WILL TAKE PLACE IN THE TABLE.
C A FINAL PASS WILL READ + REWRITE EACH RECORD WITH THE CORRECT CHAIN
C POINTERS.
C
C     DIRECTORY FILE
C 1) THE DIRECTORY FILE WILL CONTAIN A RECORD FOR EACH SCOPE(A SCOPE IS
C THE AREA WHICH THE DEFINITION OF THE TO ENCOMPASSES. A SCOPE WILL BE
C EITHER A LESSON , OR A SEGMENT.  WORD I OF EACH RECORD WILL POINT
C TO THE FIRST RECORD OF THE CHAIN FOR TO NO. I. THUS A DIRECTORY RECORD
C IS EQUIVALENT TO A SCOPE. A SCOPE WILL BE A LESSON IF LEVEL = 0, OTHER
C -WISE IT WILL BE THE ENTIRE LESSON.
C
C     THE FIRST PARAMETER, INUNIT SPECIFIES THE DEVICE FROM WHICH THE
C MBO RECORDS(80-CHAR.,EBCDIC) WILL BE READ.
C THE FIRST DIMENSION WILL INDEX BY LESSON , THE SECOND BY DECK NO.,
C THE THIRD WILL BE 1-4, WHERE WORD 1 CONTAINS THE SEGMENT NO.,
C              WORD 1 = SEGMENT NO.
C              WORD 2 = TYPE
C              WORD 3 = RECORD NO. OF FIRST QUESTION RECORD
C              WORD 4 = NUMBER OF QUESTION RECORDS
C
           COMMON/SYSTEM/NLESS,NDECK,NRES,NQUEST,NSTUD
           COMMON/FILES/IW1(2),IPRINT,IW2(2),IHEAD,IDIR,IQUEST,IW3(3)
       2   ,ISYS,IW4(3)
           INTEGER * 2  LPERM   ,   NPERM   , IDREC ,
       2   PTRS(50),      NOS(50)   , HRCORD(65) , LTABLE(4,10,40)
       3   , MRCORD(40)  ,LESSON,   SEGMNT , TC , MBO ,KEY(17)   ,
       4   SKIL1, SKIL2 , MEDIA, TYPE, QUEST , PRES(13) ,COURSE
           INTEGER * 2 QSTION(70) , DRCTRY(200), CHAIN(3,4000)
       2   , POINT1 , POINT2  , QPOINT     ,   QTO   ,    QMBO
           INTEGER RNG , DECK , RNAME(2) , HEAD , POINT
           DATA RNAME/'MBO',' '/
           EQUIVALENCE   (LESSON   , MRCORD(1) )   , (SEGMNT , MRCORD(2) ) ,
       2  (TC , MRCORD(3) ) ,   (MBC , MRCORD(4)    ), (KEY(1) , MRCORD(5))
       3   , (SKIL1, MRCORD(22) ) ,    (SKIL2, MRCORD(23) ) ,
        MIMS III SOURCE STATEMENT LISTING

      4     (MEDIA , MRCORD(24) ) , (TYPE , MRCORD(25) )  ,
      5     (QUEST , MRCORD(26)  ) , (PRES(1) , MRCORD(27) ) ,
      6     (COURSE, MRCORD(40) )  , (QTL  , QSTION(6) )  ,
      7     ( GMOD , QSTION(7)  )  , (QPOINT , QSTION(8) )
            INTEGER LIST(2)
            INTEGER DATE(6)
            INTEGER * 2 IRUN
C BEGIN BY BUILDING A TABLE FOR DIRECT ACCESS TO THE QUESTION FILE.
            IREX = 0
            READ(ISYS'2) LPERM,NPERM
            IF(LPERM.GT.0) GO TO 200
            CALL ERROR(RNAME,0,1)
            RETURN
  200       CONTINUE
            MINLES = LPERM + 1
            READ(ISYS'3) PTRS
            READ(ISYS'4) NOS
C NOW CREATE THE TABLE
            DO 500 LES = 1,LPERM
            I1 = PTRS(LES)
            I2 = PTRS(LES) + NOS(LES) - 1
            DECK = 0
            DO 400 KNO = I1,I2
C READ THE HEADER
            READ(IHEAD'KNO) HRCORD
            DECK = DECK + 1
C PLACE THE INFO IN THE TABLE
            LTABLE(1,DECK,LES) = HRCORD(3)
            LTABLE(2,DECK,LES) = HRCORD(4)
            LTABLE(3,DECK,LES) = HRCORD(8)
            LTABLE(4,DECK,LES) = HRCORD(13)
  400       CONTINUE
  500       CONTINUE
C HAVING BUILT THE TABLE WE CAN BEGIN TO READ MBC RECORDS
  800       CONTINUE
            READ(INUNIT,801,END=1600) MRCORD
  801       FORMAT( I2 , I1 , I3 , I2 , 17A2 , I1, I1 , I2 , I2 , I2 , 1X ,
         2 12A2,A1, 2X, I2 )
            IREX = IREX + 1
C HAVING READ A RECORD CHECK ITS LESSON
            IF(LESSON.GE.1.AND.LESSON.LE.LPERM) GO TO 900
            I = LESSON
            J = SEGMNT
            K = TYPE
            CALL ERROR(RNAME,0,2,IREX,I,J,K)
            GO TO 800
  900       CONTINUE
C NOW SEARCH THE TABLE
            I1 = NOS(LESSON)
            DO 1500 I = 1,I1
            IF(LTABLE(1,I,LESSON).NE.SEGMNT) GO TO 1500
            IF(LTABLE(2,I,LESSON).NE.TYPE ) GO TO 1500
C HERE WE'VE FOUND THE TEST SO CHECK THE QUESTION NUMBER
            IF(QUEST.GE.1.AND.QUEST.LE.LTABLE(4,I,LESSON))GO TO 1000
C  THE QUESTION NUMBER IS INCORRECT
            J=LTABLE(4,I,LESSON)
            K=QUEST
            CALL ERROR(RNAME,0,3,IREX,J,K)
            GO TO 800
```

```
C CHECK THE MBO AND TO NUMBERS
 1000    IF(TO.GT.0.AND.TO.LE.200.AND.MBO.GE.0) GO TO 1100
         J = TO
         K = MBO
         CALL ERROR(RNAME,0,4,IREX,J,K)
         GO TO 800
C ALL O.K. SO PLACE INFO ON DISK
 1100    RNO = ITABLE(3,1,LESSON) + QUEST - 1
         IF(LESSON.LT.MINLES) MINLES=LESSON
         READ(IQUEST'RNO) QSTION
         QTO = TO
         QMBO = MBO
         QSTION(13) = SKIL1
         QSTION(14) = SKIL2
         QSTION(15) = MEDIA
         DO 1200 J = 1,17
         QSTION(J + 40) = KEY(J)
 1200    CONTINUE
         DO 1300 J = 1,13
         QSTION(J + 57 ) = PRES(J)
 1300    CONTINUE
         WRITE(IQUEST'RNO) QSTION
         GO TO 300
 1500    CONTINUE
C IF WE FALL THROUGH HERE NO MATCH WAS FOUND
         I = LESSON
         J = SEGMNT
         K = TYPE
         CALL ERROR(RNAME,0,5,IREX,I,J,K)
         GO TO 800
C
C THIS WHOLE SCHMAGEGY IS TO DETERMINE WHICH LESSON TO BEGIN CHAINING.
C IT WILL BE EITHER THE FIRST UNCHAINED LESSON OR THE LOWEST LESSON FOR
C WHICH AN MBO WAS READ,. WHICHEVER IS LOWEST
C
C FIRST DETERMINE LAST UNCHAINED LESSON, AND LAST USED DIRECTORY RECORD
 1600    CONTINUE
         IDREC=1
         LESS = 1
         IF(LEVEL.NE.0) GO TO 2000
         DO 1700 I = 1 , LPERM
         J = PTRS(I)
         READ(IHEAD'J) HRCORD
         IF( HRCORD(7).EQ.0) GO TO 1800
         LESS = I
         IDREC = HRCORD(7)
 1700    CONTINUE
 1800    CONTINUE
C IF THIS THE LOWEST LESSON FOR WHICH MBO CARDS WERE READ WE HAVE
C THE CORRECT PARAMETERS SO WE CAN GO CHAIN
         IF(LESS.LE.MINLES) GO TO 2000
C WE'VE REPLACED OR ADDED MBO INFORMATION TO THE ALREADY CHAINED PORTION
C OF THE FILES SO WE MUST RECHAIN FROM MINLES
         LESS = MINLES
         I1 = PTRS(LESS)
         READ(IHEAD'I1) HRCORD
         IDREC = HRCORD(7)
C IF NO RECORDS WERE READ AND ALL WERE CHAINED , RETURN
 2000    CONTINUE
```

```
           IF(LESS.GT.LPERM) GO TO 7000
C
C THE NEXT AVAILABLE DIRECTORY RECORD IS IN IDREC
C THE LESSON TO CHAIN IS IN LESS
C
C CLEAR THE DIRECTORY RECORD
           DO 2100 I = 1,200
 2100      DRCTRY(I) = 0
C
C DETERMINE THE 1ST AND LAST RECORDS OF THE QUESTION FILE TO BE CHAINED
           I1 = PTRS(LESS)
           I2 = PTRS(LPERM) + NQS(LPERM) - 1
           READ(IHEAD'I1) HRCORD
           I1 = HRCORD(8)
           READ(IHEAD'I2) HRCORD
           I2 = HRCORD(8) + HRCORD(13) -1
C NOW PLACE MBO INFORMATION IN THE CHAINING AREA
C RECORD NUMBER IS POSITION IN THE FILE
           DO 2200 I = I1 , I2
           READ(IQUEST'I) QSTION
           CHAIN(1,I) = QTO
           CHAIN(2,I) = QMBO
           CHAIN(3,I) = 0
 2200      CONTINUE
C WE'RE ALL SET SO START TO CHAIN
C NOW WE START CHAINING
C
C DO LOOP FOR ALL LESSONS
 2400      CONTINUE
           DO 5500 LS = LESS,LPERM
           IFRST = PTRS(LS)
           ILAST = PTRS(LS) + NQS(LS) - 1
C DO LOOP FOR EACH HEADER
           DO 5000 HEAD = IFRST , ILAST
           READ(IHEAD'HEAD) HRCORD
C NOW WE BEGIN TO RUN THROUGH CHAIN FOR THE RECORDS SPECIFIED BY THE
C HEADER, FIRST SET UP THE DO LOOP
           IR1 = HRCORD(8)
           IR2 = HRCORD(13) + IR1 -1
           DO 4500 RNO = IR1,IR2
C IF NO TO NUMBER IGNORE RECORD
           TO = CHAIN(1,RNO)
           IF(TO.EQ.0)GO TO 4500
C BUT IF TO, THEN SEE IF FIRST OF THAT NUMBER
           MBO = CHAIN(2,RNO)
           IF(DRCTRY(TO).NE.0) GO TO 2800
C ITS THE FIRST IN THE CHAIN, SO PUT HEAD POINTER TO POINT TO IT
           DRCTRY(TO) = RNO
           GO TO 4500
 2800      CONTINUE
C ITS NOT THE FIRST, SO WE MUST SORT TO SET UP CHAIN IN MBO ORDER
C FIRST SEE IF THE NEW MBO NO IS LESS THAN THAT OF THE TOP OF THE CHAIN
           I  = DRCTRY(TO)
           IF(MBO.GE.CHAIN(2,I)) GO TO 3000
C IT IS SO PLACE,NEW RECORD AT THE TOP OF THE CHAIN
           CHAIN(3,RNO) = I
           DRCTRY(TO) = RNO
           GO TO 4500
 3000      CONTINUE
```

AIMS III SOURCE STATEMENT LISTING

```
C IT DOESN'T BELONG AT THE TOP OF THE STACK SO PLACE IT SOMEWHERE ELSE
C INITIALIZE FOR THE LOOP
         POINT1 = DRCTRY(TO)
 3400    POINT2 = CHAIN(3,POINT1)
C SEE IF THE END OF THE CHAIN
         IF(POINT2.NE.0) GO TO 3800
C YES AT END OF CHAIN SO PLACE THERE
         CHAIN(3,POINT1) = RNO
         GO TO 4500
C NOT AT END SO SEE IF IT BELONGS HERE
 3800    IF(MBO  .GE. CHAIN(2,POINT2) ) GO TO 4200
C YES IT BELONGS BETWEEN POINT1 AND POINT2
         CHAIN(3,RNO) = POINT2
         CHAIN(3,POINT1) = RNO
         GO TO 4500
 4200    CONTINUE
C NOT YET SO SEARCH FURTHER
         POINT1 = POINT2
         GO TO 3400
C THIS CLOSES THE CHAINING DO LOOP
 4500    CONTINUE
C NOW REWRITE THE HEADER WITH ITS DIRECTORY POINTER
         HRCORD(7) = IDREC
         WRITE(IHEAD'HEAD) HRCORD
C CLOSE THE LESSON DO LOOP
 5000    CONTINUE
C NOW START A NEW DIRECTORY RECORD, BUT IF NAVY ONLY AT THE END OF THE
C COURSE.
         IF (LEVEL.EQ.0) GO TO 5300
         IF(LS.NE.LPERM) GO TO 5500
 5300    CONTINUE
        WRITE(IDIR'IDREC) DRCTRY
        IDREC = IDREC + 1
        DO 5200 I = 1,200
 5200   DRCTRY(I) = 0
C WE'RE DONE , CLOSE MAJOR LOOP
 5500   CONTINUE
C FINISHED WITH THE CHAINING
C NOW MAKE A PASS THROUGH THE QUESTION FILE TO PLACE POINTERS
C IN THE RECORDS
         I1 = PTRS(LESS)
        I2 = PTRS(LPERM) + NOS(LPERM) - 1
         READ(IHEAD'I1) HRCORD
         I1 = HRCORD(8)
         READ(IHEAD'I2) HRCORD
         I2 = HRCORD(8) + HRCORD(13) - 1
C NOW PLACE POINTERS IN THE QUESTION FILE
         DO 5700 RNO = I1 , I2
         READ(IQUEST'RNO) QSTION
         QPOINT = CHAIN(3,RNO)
         WRITE(IQUEST'RNO) QSTION
 5700    CONTINUE
C  REWRITE NO. OF DIRECTORY RECORDS
         IDREC = IDREC - 1
         WRITE(ISYS'6) IDREC
C LIST THE FILE IN MBO ORDER
         ENTRY MBOLST
 7000    CONTINUE
         CALL INFO(DATE)
```

```
      AIMS III SOURCE STATEMENT LISTING

          READ(ISYS'6) IDREC
          IPAGE = 0
          IREX = 0
          IF(IDREC.LE.0) GO TO 8600
C LOOP FOR EACH DIRECTORY RECORD
          DO 8500 IDN = 1, IDREC
          READ(IDIR'IDN) DRCTRY
          IW = 0
C LOOP FOR EACH TG
          DO 8000 TG = 1,99
          RNO = DRCTRY(TG)
 7300     IF(RNO.LE.0) GO TO 8000
          IF(IW.GT.0) GO TO 7500
          IPAGE = IPAGE + 1
          WRITE(IPRINT,7001) DATE,IDN,IPAGE
          WRITE(IPRINT,9302)
          IW = 10
 7500     CONTINUE
          IREX = IREX + 1
          READ(IQUEST'RNO) QSTION
          CALL LSTCDE(QSTION(12),LIST)
          WRITE(IPRINT,9001) (QSTION(L),L=1,3) , (QSTION(L),L=5,7)   ,
     2    LIST ,  (QSTION(L),L=13,15) , (QSTION(L),L=21,70)
          RNO= QPOINT
          IW = IW - 1
          GO TO 7300
 8000     CONTINUE
 8500     CONTINUE
 8600     CONTINUE
          WRITE(IPRINT,9701) IREX
          RETURN
C LIST THE FILE IN QUESTION ORDER
          ENTRY QLIST
 9000     CALL INFO(DATE)
C FIND NO. OF QUESTION RECORDS
          READ(ISYS'5) IQNUM
          IF(IQNUM.LE.0) GO TO 9700
C READ AND WRITE EACH RECORD
          IW = 0
          IPAGE = 0
          DO 9500 RNO = 1, IQNUM
          IF(IW.GT.0) GO TO 9300
          IPAGE = IPAGE + 1
          WRITE(IPRINT,9301) DATE, IPAGE
          WRITE(IPRINT,9302)
          IW = 10
 9300     CONTINUE
          READ(IQUEST'RNO) QSTION
          CALL LSTCDE(QSTION(12),LIST)
          WRITE(IPRINT,9001) (QSTION(L),L=1,3) , (QSTION(L),L=5,7)   ,
     2    LIST ,  (QSTION(L),L=13,15) , (QSTION(L),L=21,70)
          IW = IW - 1
 9500     CONTINUE
 9700     CONTINUE
          WRITE(IPRINT,9701) IQNUM
          RETURN
 7001     FORMAT(1H1 ,  7X , '***  ','AIMS MBO LISTING',
     2        '  ***',  56X    ,'JOB', 2X , 2A4 ,   2X ,
     3   2A4   ,  1X,  2A4  /     13X  ,'SCOPE NUMBER',  I4  ,
     4          62X  ,'PAGE'  ,  I4 ///)
 9001     FORMAT(2X , I2 , 3X , I2 , 2X , I1 , 2X ,I2 , 1X , I2 , 1X  ,
     2   I2, 1X , 2A4 , 1X , I1 ,  3X ,  I1 ,  3X , I2 ,  2X , I3 ,
     3   2X , I3 , 2X  , I3 ,  2X  ,   I3 ,  3X ,  I3  , 3X  , I3  , 3X,
     4   I3 , 2X , I3 , 1X ,  I3  ,  11(1X, I3 )  /1H0 ,17A2, 6X  ,
     5   12A2,A1   / )
 9302     FORMAT( 1X , 'LESS' , 1X , 'SEG' ,  1X , '}P' , 1X , 'QST' ,
     2    1X , 'TO', 1X , 'EQ' , 1X , 'CORRECT',  1X , 'SKL' , 1X ,
     3    'SKL',  1X , 'MDIA' , 1X , 'PCT.' ,   1X ,  'VALD' , 1X ,'STD',
     4    1X  ,  'AVG' , 3X  , 'AVG', 3X ,'N' , 5X , 'N' , 5X  , 'HI' ,
     5    2X ,'LOW' , 1X , 'RESPONSE COUNTERS'   /
     6    23X , 'ANSWER' , 3X ,'1',  5X , '2' , 7X , 'HITE' ,  6X  ,
     7    'DEV' , 1X , 'RT' ,  1X , 'GP' , 1X  , 'WG',  1X , 'UP' , 1X ,
     8    'WT' ,   1X , 'GP'  , 1X , 'WG' , 1X , 'UP' ,  1X , 'GRP' ,
     9    1X ,  'GRP' ,  1X ,  'BLK', 1X , 'A' , 3X ,  'B' ,  3X , 'C' ,
     A    3X , 'D' , 3X ,  'E' , 3X , 'F' , 3X , 'G' , 3X ,  'H' ,  3X ,
     B    'I' , 3X  ,  'J'  /  /   )
 9301     FORMAT(1H1 , 7X , '***  ', 'AIMS QUESTION LISTING' ,
     2         '  ***',  51 X  , 'JOB', 2X,  2A4 ,  2X  ,
     3    2A4 ,  1X ,  2A4   /   91X ,    'PAGE' , I4  ///)
 9701     FORMAT(1X,    3(/) ,  15X               ,
     2    'THERE ARE' , I4 ,  ' RECORDS IN THIS FILE' )
          END
/*
 PHASE AIMSPRC1,AIMSHEAD
// EXEC FFORTRAN
          SUBROUTINE PROCES(LESSON)
          COMMON/FILES/IN(15)
          COMMON/SYSTEM/IL(5)
          INTEGER DECKS
          INTEGER * 2 ANSWER(10,185),SECOND(10,185),QSTION(18,48,10),
     2    SAMPLE(60),SDEV(60),MEAN(60),MIN(60),MAX(60),REPLY(55,10),
     3    HEADER(65,10)
          EQUIVALENCE(REPLY(1,1),SECOND(1,1)),
     2    (HEADER(1,1),SECOND(10,60) )
          CALL OPSYS('LOAD','AIMSPRC2')
          CALL OPSYS('LOAD','AIMSTAPE')
          CALL PROC1(LESSON,DECKS,ANSWER,QSTION,REPLY,HEADER,
     2 SAMPLE,SDEV,MEAN,MIN,MAX )
          CALL OPSYS('LOAD','AIMSPRC3')
          CALL PROC2(LESSON,DECKS,ANSWER,SECOND,QSTION,
     2 SAMPLE,SDEV,MEAN,MIN,MAX )
          RETURN
          END
/*
 PHASE AIMSPRC2,*
// EXEC FFORTRAN
          SUBROUTINE PROC1(LESSON,DECKS, ANSWER , QSTION, REPLY,HEADER ,
     2    SAMPLE , SDEV, MEAN , MIN, MAX )
C
C    THIS, THE FIRST OF THE TWO AIMS PROCESS ROUTINES, DOES THE ACTUAL
C GRADING, USING BOTH THE STUDENT'S RESPONSES AND THE CORRECT ANSWERS.
C THIS ROUTINE WILL CALCULATE 1 GRADE FOR EACH DECK IN THE LESSON. AS
C WELL AS THE MINIMUM, MAXIMUM, MEAN, STD. DEV. OF EACH OF THE GRADES.
C     IN ADDITION THE VALIDITY, AND RESPONSE COUNTS FOR THE QUESTIONS
C ARE ALL DONE IN CORE IN THIS ROUTINE, AND PASSED TO THE 2ND PROCESS
C ROUTINE AS PARAMETERS. THIS ROUTINE WILL CREATE A SCRATCH FILE OF THE
C RESPONSE DATA, WHICH WILL BE USED IN THE 2ND OF THE PROCESS ROUTINES
C TO GET COUNTS, FOR EACH QUEST, OF THE STUDENTS ABOVE THE MEAN AND GET
```

```
C IT RIGHT, AND THOSE BELOW WHO GOT IT RIGHT.
C     THE FIRST SECTION WILL SET UP FOR I/O, THAT IS FIND THE CORRECT
C LESSON ON THE TAPE, AND READ THE APPROPRIATE HEADERS.
          COMMON/SYSTEM/NLESS,NDECK,NREC,NQUEST,NSTUD
          COMMON/FILES/IW1(2),IPRINT,IRESP,ISCRCH, IHEAD,IW2(2),
     2      ISTDNT, ISCORE ,  IAS ,  ISYS , IW4(3)
          INTEGER *2 LPERM ,NOS(50),  PTRS(50) , HEADER(65,10) ,ROSTER,
     2    SUPER,COUNT(11), SCRTCH(65)
         INTEGER PNAME(2),FLAG  , DECKS , TEST  , TYPE , STDENT, NQST,
     2    SELECT, NSELCT , RITE , WRONG , WRONG2 , QST
          DATA PNAME/'PROC','ESS'/
          INTEGER * 2 REPLY(65,10) , QSTION(17,48,10) , ANSWER(10,150)  ,
     2    MEAN(60) , SDEV(60) ,SAMPLE(60),MAX(60) , MIN(60), ANS, RSP
          REAL * 8       A , B , C , SUMSQ(10) , SUM(10)
          INTEGER ENDFIL , COUNTR, LRI , POINTR , GRD
          INTEGER * 2 MISSED
          DATA MISSED/-999/
          IROUND(E) = E + 0.5
C
C SEE IF THIS LESSON IS IN PERMANENT
C
          READ(ISYS'2) LPERM
          IF(LESSON.GT.0.AND.LESSON.LE.LPERM) GO TO 200
C ILLEGAL, SO CAN'T BE GRADED
          I = LESSON
          J = LPERM
          CALL ERROR(PNAME,0,1,I,J)
          RETURN
C
C THE NUMBER IS LEGAL SO SEARCH THE TAPE FOR THE CORRECT RECORD
  200     CONTINUE
          CALL TAPSVC(LESSON,FLAG)
C  IF NOT FOUND ALREADY AN ERROR MESSAGE SO JUST RETURN
          IF(FLAG.NE.0) RETURN
C
C NOW READ THE MAP OF THE HEADER FILE , AND THEN THE HEADERS
C
          READ(ISYS'3) PTRS
          READ(ISYS'4) NOS
          DECKS = NOS(LESSON )
          I1 = PTRS(LESSON)
          I2 = I1 + NOS(LESSON) - 1
          J = 0
          DO 400 I = I1 , I2
          J = J + 1
          READ(IHEAD'I) (HEADER(K,J),K =1,65)
  400     CONTINUE
C
C WE WILL NOW INITIALIZE VARIABLES FOR ACCUMULATION, ETC.
C
C INITIALIZE QUESTION COUNTERS
          DO  600 K = 1, 10
          DO  600 J = 1, 48
          DO  600 I = 1,17
          QSTION(I,J,K) = 0
  600     CONTINUE
          DO 700 I = 1 , 10
          SUMSQ(I) = 0.0
          SUM(I) = 0.0
```

```
      AIMS III SOURCE STATEMENT LISTING

  700     CONTINUE
          DO  800 I = 1,60
          MAX(I) =  -1000
          MIN(I) =  10000
          SDEV(I)= 0
          SAMPLE(I) = 0
          MEAN(I) = 0
  800     CONTINUE
          ENDFIL = 0
C
C READ THE NUMBER OF STUDENTS
C
          READ(ISYS'1) ROSTER
          IF(ROSTER.GT.0) GO TO 850
          CALL ERROR(PNAME,0,6)
          RETURN
  850     CONTINUE
C THIS DO LOOP IS FOR ALL STUDENTS
C
          DO 6500 STDENT = 1, ROSTER
          DO 875 I = 1,10
          ANSWER(I,STDENT) = 0
  875     CONTINUE
          COUNTR= 0
          LRI = 0
          POINTR = 1
          DO 6250 TEST = 1 , DECKS
C GET SOME DATA IF ANY
C IF POINTR  IS LE COUNTRS HAVE DATA
  900     IF( POINTR. LE . COUNTR) GO TO 1500
C IF NOT GET DATA IF POSSIBLE
C FIRST CHECK FOR EOF, 6000 IS MISSING DATA
          IF(ENDFIL . NE. 0) GO TO 6000
C CHECK THE LAST RECORD INDICATOR (LRI) TO SEE IF THERE ARE MORE
C RECORDS FOR THTS STUDENT
          IF( LRI . NE. 0 ) GO TO 6000
          CALL TPDATA( STDENT , FLAG , LRI ,COUNTR, REPLY)
C IF FLAG = 0 THE DATA WAS OBTAINED
          IF(FLAG . NE . 0 ) GO TO 1000
          POINTR = 1
          GO TO  1500
 1000     CONTINUE
C NO DATA WAS OBTAINED
C FLAG = 1 , NO DATA
C FLAG = 2 , END OF FILE
C FLAG = 3 , SORT ERROR, TPDATA GAVE ERROR MESSAGE , SO JUST RETURN
          GO TO(6000 ,1400 , 1200) , FLAG
 1200     CONTINUE
          RETURN
 1400     CONTINUE
          ENDFIL = 1
          GO TO 6000
 1500     CONTINUE
C WE HAVE SOME DATA, BUT IS IT THE RIGHT DECK
          IF( REPLY(3,POINTR) - HEADER(3,TEST) ) 1700 , 1600, 6000
C RIGHT SEGMENT , WHAT ABOUT TYPE
 1600     CONTINUE
          IF( REPLY(4,POINTR) - HEADER(4,TEST) ) 1700, 1800, 6000
C ILLEGAL DATA RECORD
```

```
      AIMS III SOURCE STATEMENT LISTING

 1700     CONTINUE
          I = REPLY(1,POINTR)
          J = REPLY(2,POINTR)
          K = REPLY(3,POINTR)
          L = REPLY(4,POINTR)
          CALL ERROR(PNAME,0,5,I,J,K,L)
          POINTR = POINTR + 1
          GO TO 900
C
C WE'VE GOT THE CORRECT DATA RECORD, SO CALCULATE A GRADE
 1800     CONTINUE
C WE NOW ZAP THROUGH A TEST AND GATHER BASIC INFORMATION
          TYPE = HEADER(4,TEST)
          NOQST = HEADER(13,TEST)
          SELECT = 0
          RSELCT = 0
          RITE  = 0
          NOGRD = 0
          DO 2000  I  = 1 , 17
          SCRTCH(I) = REPLY(I,POINTR)
 2000     CONTINUE
          DO 3000 QST = 1, NOQST
          QNO = QST + 17
          ANS = HEADER(QNO,TEST)
          RSP = REPLY(QNO,POINTR)
          NO = IPASS(ANS,RSP,NOPCH,COUNT)
C COUNT THE RESPONSES
          DO 2100 I = 1, 11
          QSTION(I,QST,TEST) = QSTION(I,QST,TEST) + COUNT(I)
 2100     CONTINUE
C IF ANSWER IS BLANK'DON'T GRADE THE QUESTION
          IF( ANS .EQ . 1)  GO TO 2600
C SEE IF HE GOT A  RIGHT RESPONSE
          IF( NO. EQ .0 ) GO TO 2600
          RSELCT = RSELCT + 1
          IF(NO. EQ. NOPCH) RITE = RITE + 1
 2500     CONTINUE
          SELECT = SELECT + ( NOPCH - NO)
          IF(RSP.NE.1) NOGRD = NOGRD + 1
 2600     CONTINUE
          SCRTCH(QNO) = NO
 3000     CONTINUE
C
C WE NOW HAVE A GRADE , SO PERFORM FURTHER CALCULATIONS
C
C  HERE WE HAVE ALL THE INFORMATION NECESSARY TO CALCULATE
C  A GRADE AND SO WE SHALL DO SO
C  EACH STATEMENT IS FOR A DIFFT ALGORITHM AND THERE ALSO SHALL
C  BE DIFFERENT ALGORITHMS FOR NAVY AND HEW
       GO TO(3200,3400,3600,3800,3400,4200,4400,4600,4800),TYPE
 3400     CONTINUE
          C = RITE
          Q = HEADER(12,TEST)
          GRADE =  (C / Q )  * 100.0
          GO TO 5000
 3600     CONTINUE
          GRADE = 0.0
          DO 3700 QST = 1 , NOQST
          QNO = QST + 17
```

AIMS III SOURCE STATEMENT LISTING

```
          RSP = REPLY(QNO,POINTR)
          IF(RSP.NE.3) GO TO 3640
          GRADE = GRADE +  10.0
          GO TO 3700
  3640    IF(RSP.NE.5) GO TO 3660
          GRADE = GRADE + 8.0
          GO TO 3700
  3660    IF(RSP.NE.9) GO TO 3680
          GRADE = GRADE + 6.0
          GO TO 3700
  3680    IF(RSP.EQ.17) GRADE = GRADE + 3.0
  3700    CONTINUE
          GRADE = GRADE / ( HEADER(12, TEST ) ) * 10.0
          GO TO 5000
  3800    CONTINUE
          C = NSELCT
          Q = HEADER(12,TEST)
          H = SELECT
          P = HEADER(11,TEST) - Q
          X = Q * ( H / P )
        . GRADE = ( C - X ) / Q * 100.0
          GO TO 5000
  3200    CONTINUE
  4000    CONTINUE
  4200    CONTINUE
  4400    CONTINUE
  4600    CONTINUE
  4800    CONTINUE
          GRADE = 0
  5000    CONTINUE
          INTR =  IROUND(GRADE)
          IF(INTR .GT. MAX(TEST) ) MAX(TEST) = INTR
          IF(INTR .LT. MIN(TEST) ) MIN(TEST) = INTR
          SAMPLE(TEST) = SAMPLE(TEST) + 1
          SUM( TEST ) = SUM( TEST) + GRADE
          SUMSQ( TEST) = SUMSQ( TEST) + ( GRADE * GRADE )
          ANSWER(TEST,STDENT) = INTR
          SCRTCH(11) = INTR
          IREC = (( STDENT - 1) * DECKS)  + TEST
          WRITE(ISCRCH'IREC) SCRTCH
C NOWGET COUNTS AND AVGS FOR QSTION VALIDITY
          DO 5500 QST = 1, NOQST
          QNO = QST + 17
          IF(SCRTCH(QNO).LE.0)GO TO 5200
          QSTION(12,QST,TEST) = QSTION(12,QST,TEST) + INTR
          QSTION(14,QST,TEST) = QSTION(14,QST,TEST) + 1
          GO TO 5500
  5200    CONTINUE
C   NO HE'S IN THE WRONG GROUP
          QSTION(13,QST,TEST) = QSTION(13,QST,TEST) + INTR
          QSTION(15,QST,TEST) = QSTION(15,QST,TEST) + 1
  5500    CONTINUE
          POINTR = POINTR + 1
          IF(TEST.NE.DECKS) GO TO 6250
          IF(POINTR.GT.COUNTR) GO TO 6250
          DO 5750 M = POINTR,COUNTR
          I = REPLY(1,M)
          J = REPLY(2,M)
          K = REPLY(3,M)
```

FILMED FROM BEST AVAILABLE COPY

```
          GIPS III SOURCE STATEMENT LISTING

         L = REPLY(4,M)
         CALL ERROR(PNAME,0,5,I,J,K,L)
 5750    CONTINUE
         GO TO 6250
C MISSING DATA
 6000    CONTINUE
         ANSWER(TEST,STUDENT) = -999
         IREC = (( STUDENT - 1) * DECKS)  + TEST
         WRITE(ISCRCH'IREC) MISSED
 6250    CONTINUE
 6500    CONTINUE
         DO 8000 TEST = 1, DECKS
         N = SAMPLE(TEST)
         IF(N.LE.0) GO TO 7500
         A =  SUM(TEST)
         G =  A / N
         MEAN(TEST) = IROUND(G)
         IF(N.GT.1) GO TO 7000
         SDEV(TEST) = 0
         GO TO 8000
 7000    CONTINUE
C CALCULATE STANDARD DEVIATIONS
         B = A * A
         C = N * SUMSQ(TEST)
         S = N * (N- 1)
        SS = ( C - B)  / S
        SD = SQRT(SS)
        SDEV(TEST) = IROUND(SD)
        GO TO 8000
 7500   CONTINUE
        MEAN(TEST) = -999
        SDEV(TEST) = -999
        MAX(TEST)  = -999
        MIN(TEST)  = -999
 8000   CONTINUE
        RETURN
        END
/*
// ASSGN SYSSLB,X'191'
// EXEC ASSEMBLY
IPASS    START
         CUME  ENTER
CORECT   EQU   4
STUDNT   EQU   6
ENTER    L     3,0(0,1)                 1ST PAR IS CORECT ANS
         LH    5,0(0,3)                 PLACE CORECT ANS IN GPR5
         L     3,4(0,1)                 2ND PAR IS STUDENT'S RESPONSE
         LH    7,0(0,3)                 PLACE STUDENT ANS IN GPR7
         NR    5,7                      CHECK CORRECT BITS
         L     11,12(0,1)               LOAD GPR11 WITH ADDR OF COUNTS
*                                       COUNTS ARE 4TH PARAMETER
         USING CNTAR,11                 TELL ASSEMBLER
         MVI   BLANK,X'00'              CLEAR COUNT AREA
         MVC   BLANK+1(21),BLANK
         SR    0,0                      CLEAR GPR0 FOR TOTAL CORRECT
         SR    3,3                      CLEAR GPR3 FOR TOTAL
         SLL   5,21                     LEFT SHIFT TO SET UP FOR COUNTS
         SLL   7,21
*                                       THIS LOOP BEGINS THE COUNTING
```

```
        AIMS III SOURCE STATEMENT LISTING

  *                                             SET UP FOR THE LOOP
           LA     10,18(0,0)                    THIS IS OUR INDEX GPR
           L      8,=F'-2'                      THIS IS THE INCREMENT
           L      9,=F'-1'                      THIS IS THE COMPARAND
 LOOP      SR     CORECT,CORECT                 CLEAR THE HI ORDER GPR'S
           SR     STUDNT,STUDNT
           SLDL   CORECT,1                      PLACE A BIT IN THE GPR
           SLDL   STUDNT,1                      ALSO FOR HIS RESPONSE
           AR     6,CORECT                      COUNT THE CORRECT BITS
           AR     3,STUDNT                      COUNT THE STUDENT RESPONSE
           STH    STUDNT,COUNTS(10)             AND COUNT THE RESPONSE
           BXH    10,8,LOOP                     FINISH LOOP
           C      3,=F'0'
           BC     X'2',SOME                     IF NONE OTHERS,THEN A BLANK
           L      4,=F'1'
           STH    4,BLANK                       COUNT THE BLANK
 SOME      L      4,8(0,1)                      GET ADR OF 3RD PARAMETER
           STH    3,0(0,4)                      PLACE RESPONSE CNT IN 3RD PAR
           HOME
 CNTAR     DSECT
 BLANK     DS     1H
 COUNTS    DS     10H
           END
 /*
  PHASE AIMSTAPE,*
 // EXEC FFORTRAN
           SUBROUTINE TAPSVC(LESSON,FLAG)
 C
 C THE MAIN ENTRY POINT TO THIS ROUTINE WILL SEARCH THE TAPE FOR
 C THE REQUESTED LESSON, IF FOUND, FLAG WILL BE RETURNED AS 0
 C OTHERWIDE A 1 WILL INDICATE THAT NO DATA WAS FOUND FOR THE LESSON, OR
 C A 2 WILL INDICATE AN UNSORTED TAPE
           COMMON/FILES/IW(3),ITAPE,IW1(11)
           INTEGER ENDFIL, RDFLAG , RCOUNT, FLAG, COUNT, STUDNT
           INTEGER * 2  SORT(4), RSPONS(65) , REPLY(65,10)
           INTEGER PNAME(2)
           DATA PNAME/'PROC','ESS'/
 C OPEN THE FILE AND INITIALIZE
           REWIND ITAPE
           ENDFIL = 0
           DO 100 I = 1,4
           SORT(I) = 0
   100     CONTINUE
 C SET  THE RDFLAG TO OFF
           RDFLAG = 0
           RCOUNT = 0
           LESS = LESSON
           REWIND ITAPE
   500     READ(ITAPE,END=1000) RSPONS
           RCOUNT = RCOUNT + 1
 C BEGIN TO CHECK FOR SORT ERROR
           DO 700 I = 1 , 4
           IF(SORT(I) - RSPONS(I) )  800 ,700, 600
   600     CONTINUE
 C THIS IS A SORT ERROR
           J = SORT( I )
           K = RSPONS( I )
           CALL ERROR(PNAME,0,2,RCOUNT,1,J,K)
           FLAG = 2
```

```
            RETURN
  700       CONTINUE
C HERE WE HAVE 2 IDENTICAL RECORDS, THIS IS A BADNESS
            I = RSPONS(2)
            J = RSPONS(3)
            K = RSPONS(4)
            CALL ERROR(PNAME,0,3,RCOUNT,I,J,K)
            FLAG = 2
            RETURN
  800       CONTINUE
C NO SORT ERROR SAVE SORT PARAMETERS
            DO 900  J = 1,4
            SORT(J) = RSPONS(J)
  900       CONTINUE
C NOW CHECK THE LESSON
            IF(RSPONS(1) - LESSON)  500,950,1000
C THIS IS THE RIGHT LESSON
  950       CONTINUE
C BUT  IF ITS A HEADER SKIP IT
            IF(RSPONS(2).EQ.0) GO TO 500
C WE'RE HERE SO CAN RETURN
            FLAG = 0
            RETURN
C THE LESSON ISN'T ON THIS TAPE
 1000       CONTINUE
            ENDFIL = 1
            REWIND ITAPE
            I= LESSON
            J = SORT(1)
            CALL ERROR(PNAME,0,4,I,J)
            FLAG = 1
            RETURN
C THIS ENTRY SEARCHES FOR A GIVEN STUDENT'S DATA
C IF NONE IS FOUND FLAG WILL RETURN A 1, ON END OF FILE ( THAT IS THE
C FIRST CALL WITH NO DATA RETURNED FLAG WILL RETURN A 2
C
            ENTRY TPDATA(STUDNT,FLAG,LCI,COUNT,REPLY)
C SET COUNT TO 0 , I. E. NO DATA
            COUNT = 0
            LCI = 0
            FLAG = 0
            IF(ENDFIL.NE.0) GO TO 9000
C WE CAN READ SHOULD WE
 2000       CONTINUE
            IF(RDFLAG.EQ.0) GO TO 3000
            READ(ITAPE,END=8000) RSPONS
            IF(RSPONS(1).NE.LESS) GO TO 8000
            RCOUNT = RCOUNT + 1
C WE'VE READ A RECORD CHECK THE SORT
            DO 2200 I = 1,4
            IF(SORT(I) - RSPONS(I) ) 2300 , 2200, 2100
 2100       CONTINUE
C SORT ERROR
            J = SORT( I)
            K = RSPONS(I)
            CALL ERROR(PNAME,0,2,RCOUNT,I,J,K)
            FLAG = 3
            RETURN
 2200       CONTINUE
```

```
C IDENTICAL RECORDS
          I = RSPONS(2)
          J = RSPONS(3)
          K = RSPONS(4)
          CALL ERROR(PNAME,0,3,RCOUNT,I,J,K)
          FLAG = 3
          RETURN
C SORT IS O.K.
 2300     CONTINUE
          DO 2400 J = 1, 4
          SORT(J) = RSPONS(J)
 2400     CONTINUE
C HERE WE'VE GOT DATA, FROM READING, OR LAST TIME
 3000     CONTINUE
          RDFLAG = 1
          IF( RSPONS(2) - STUDNT) 3200, 4000, 3500
C WE MUST SKIP, BUT NOTE THE FACT
 3200     I= RSPONS(2)
          J = RSPONS(3)
          K = RSPONS(4)
          L= RSPONS(1)
          CALL ERROR(PNAME,0,5,L,I,J,K)
          GO TO 2000
C NO MORE DATA FOR STUDENT
 3500     CONTINUE
          RDFLAG = 0
          LCI = 1
C WAS THERE ANY DATA BEFORE
          IF(COUNT.NE.0) RETURN
C NO SO STUDENT HAD NO DATA
          FLAG = 1
          RETURN
C THIS DATA BELONGS TO THIS STUDENT
C BUT IS THIS THE 11TH RECORD
 4000     CONTINUE
          IF(COUNT.LT.10) GO TO 4500
C    YES
          LCI = 0
          RDFLAG = 0
          RETURN
C NO, SO PLACE DATA IN ANS
 4500     COUNT = COUNT + 1
          DO 5000 I = 1, 65
          REPLY(I,COUNT) = RSPONS(I)
 5000     CONTINUE
C AND GO READ AGAIN
          GO TO 2000
C THERE IS NO MORE DATA
 8000     CONTINUE
          ENDFIL = 1
          IF(COUNT.EQ.0) GO TO 9000
          LCI = 1
          REWIND ITAPE
          RETURN
C WE WILL NOT READ AGAIN
 9000     CONTINUE
          LCI = 1
          FLAG = 2
          REWIND ITAPE
```

```
     RETURN
     END
/*
 PHASE AIMSPRC3,AIMSPRC2
// EXEC FFORTRAN
     SUBROUTINE PROC2(LESSON,DECKS, ANSWER , SECOND , QSTION ,
   2 SAMPLE , SDEV, MEAN , MIN, MAX )
     REAL * 8 SUM(20) , SUMSQ(20) , A , B , C
     INTEGER * 2  ANSWER(10,150) , SECOND(10,150) , QSTION(10,48,10)
   2 , SAMPLE(60) , SDEV(60) , MEAN(60) , MIN(60) , MAX(60)
     INTEGER * 2 RECORD(65) , TABLE(2,10) ,PTRS(50) , IDENT(5)
   COMMON/SYSTEM/NLESS,NDECK,NREX,NQUEST,NSTUD
     COMMON/FILES/ IW1(4), ISCRCH , IHEAD, IW2 , IQUEST ,
   2 ISTDNT , ISCORE , IW3, ISYS , IW4(3)
     INTEGER * 2  CAPIN , ROSTER
     EQUIVALENCE(CAPIN,RECORD(25))
     INTEGER DECKS , STDENT , TEST , QNO, QST ,AVG , AVG2
C FIRST BUILD A TABLE (FROM THE HEADER FILE)TO DESCRIBE THE
C QUESTION FILE
C
     DO 250 I = 1,20
     SUM(I) = 0.0
     SUMSQ(I) = 0.0
 250 CONTINUE
     READ(ISYS'3) PTRS
     J = PTRS(LESSON)
     DO 500 I = 1 , DECKS
     READ(IHEAD'J) RECORD
     TABLE(1,I) = RECORD(6)
     TABLE(2,I) = RECORD(15)
     J = J + 1
 500 CONTINUE
C FIND THE NUMBER OF STUDENTS
     READ(ISYS'1) ROSTER
C  TWO   PASSES WILL BE NECESSARY TO OBTAIN ALL GRADES AS THE LAST
C  USES   MEANS FROM THE PREVIOUS PASS
C
C IN PASS 1 WE WILL 1)  COLLECT INFORMATION ON THE CAPABILITY INDICES
C                       FROM THE DISK
C                   2)  PERFORM COUNTS FOR HI AND LO GROUPS FOR QUESTION
C                       COUNTERS ( ALSO FROM DISK )
C                   3)  CALCULATE THE COMBINED SECONDARY GRADES,
C                       PERFORMANCE INDEX AND NET ACHEIVEMENT INDEX
C PASS 1
     DO 3000 STDENT = 1 , ROSTER
C FIRST GET THE CAPABILITY INDEX
     DO 750 I = 1,10
     SECOND(I,STDENT) = 0
 750 CONTINUE
     READ(ISTDNT'STDENT) RECORD
     SECOND(1,STDENT) = CAPIN
     IF(CAPIN.EQ.-999) GO TO 1000
     SAMPLE(11) = SAMPLE(11) + 1
     SUM(11) = SUM(11) + CAPIN
     SUMSQ(11) = SUMSQ(11) + (CAPIN * CAPIN )
     IF(CAPIN.GT.MAX(11) ) MAX(11) = CAPIN
     IF(CAPIN.LT.MIN(11) ) MIN(11) = CAPIN
C NOW CALCULATE THE COMBINED GRADES
C FIRST IS NET ACHEIVEMENT INDEX
```

```
        AIMS III SOURCE STATEMENT LISTING
  1000     CONTINUE
        IF(ANSWER(1,STDENT).NE.-999.AND.ANSWER(2,STDENT).NE.-999)GO TO1200
           SECOND(2,STDENT) = -999
           SECOND(3,STDENT) = -999
           GO TO 1500
  1200     A = ANSWER(1,STDENT)
           GRADE     =    (A + ANSWER(2,STDENT)) / 2.0
           INTER  = IROUND(GRADE)
           SECOND(2,STDENT) = INTER
           SAMPLE(12) = SAMPLE(12) + 1
           SUM(12) = SUM(12) + GRADE
           SUMSQ(12) = SUMSQ(12) + (GRADE * GRADE)
           IF(INTER.LT.MIN(12)) MIN(12) = INTER
           IF(INTER.GT.MAX(12)) MAX(12) = INTER
 C NOW CALCULATE ABSOLUT ACHEIVEMENT DEVIATION
           IF(CAPIN.NE.-999) GO TO 1400
           SECOND(3,STDENT) = -999
           GO TO 1500
  1400     CONTINUE
           GRADE = GRADE - CAPIN
           INTER = IROUND(GRADE)
           SECOND(3,STDENT) = INTER
           SAMPLE(13) = SAMPLE(13) + 1
           SUM(13) = SUM(13) + GRADE
           SUMSQ(13) = SUMSQ(13) + (GRADE * GRADE)
           IF(INTER.LT.MIN(13) ) MIN(13) = INTER
           IF(INTER.GT.MAX(13) ) MAX(13) = INTER
  1500     CONTINUE
 C NOW CALCULATE PERFORMANCE INDEX
        IF(ANSWER(3,STDENT).NE.-999.AND.ANSWER(4,STDENT).NE.-999)GOTO 1700
           SECOND(5,STDENT) = -999
           SECOND(6,STDENT) = -999
           GO TO 2000
  1700     CONTINUE
           A = ANSWER(3,STDENT)
           GRADE =((3.0 * A) + ANSWER(4,STDENT)) / 4.0
           INTER = IROUND(GRADE)
           SECOND(5,STDENT) = INTER
           SAMPLE(15) = SAMPLE(15) + 1
           SUM(15) = SUM(15) + GRADE
           SUMSQ(15) = SUMSQ(15) + (GRADE * GRADE)
           IF(INTER.LT.MIN(15) ) MIN(15) = INTER
           IF(INTER.GT.MAX(15) ) MAX(15) = INTER
 C NOW CALCULATE ABSOLUTE PERFORMANCE DEVIATION
           IF(CAPIN.NE.-999) GO TO 1900
           SECOND(6,STDENT) = -999
           GO TO 2000
  1900     CONTINUE
           GRADE = GRADE - CAPIN
           INTER = IROUND(GRADE)
           SECOND(6,STDENT) = INTER
           SAMPLE(16) = SAMPLE(16) + 1
           SUM(16) = SUM(16) + GRADE
           SUMSQ(16) = SUMSQ(16) + (GRADE * GRADE)
           IF(INTER.LT.MIN(16) ) MIN(16) = INTER
           IF(INTER.GT.MAX(16) ) MAX(16) = INTER
  2000     CONTINUE
 C NOW WE'LL ZAP THROUGH THE PRIMARY GRADES IF THE STUDENT'S SCORE
 C IS ABOVE OR = TO THE MEAN , WE'LL COUNT HIM IN THE RIGHT OR WRONG
```

```
      WIPS III SOURCE STATEMENT LISTING

C QUESTION COUNTERS
          DO 2600 TEST = 1 , DECKS
          IF(ANSWER(TEST,STDENT).LT. EAN(TEST) ) GO TO 2600
          IREC = ((STDENT - 1) * DECKS) + TEST
          READ((SCRCH'IREC) RECORD
          NUM = TABLE(2,TEST)
          DO 2500 QNO = 1,NUM
          QST = QNO + 17
          IF(RECORD(QST).LT.1) GO TO 2250
          QSTION(16,QNO,TEST) = QSTION(16,QNO,TEST) + 1
          GO TO 2500
 2250     CONTINUE
          QSTION(17,QNO,TEST) = QSTION(17,QNO,TEST) + 1
 2500     CONTINUE
 2600     CONTINUE
 3000     CONTINUL
C NOW PREPATORY TO PASS 2 WE'LL CALCULATE THE MEAN  OF THE
C ABSOLUTE ACHEIVEMENT DEVIATION
          IF (SAMPLE(13).GT.0) GO TO 4000
C NO ONE HAS ABS ACHV DEV
          DO 3500 I = 1,ROSTER
          SECOND(4,I) = - 888
 3500     CONTINUE
C NOW GO ON TO REL PREF DEV
          GO TO 5100
 4000     CONTINUE
          GRADE = SUM(13) / SAMPLE(13)
          AVG  = IROUND(GRADE)
C NOW CALCULATE RELATIVE ACHIEVEMENT DEVIATIONS
          DO 5000 STDENT = 1,ROSTER
          IF(SECOND(3,STDENT).NE.-999) GO TO 4500
          SECOND(4,STDENT) = -999
          GO TO 5000
 4500     CONTINUE
          GRADE = SECOND(3,STDENT) - AVG
          INTER = IROUND(GRADE)
          SECOND(4,STDENT) = INTER
          SAMPLE(14) = SAMPLE(14) + 1
          SUM(14) = SUM(14) + GRADE
          SUMSQ(14) = SUMSQ(14) + (GRADE * GRADE)
          IF(INTER.LT.MIN(14) ) MIN(14) = INTER
          IF(INTER.GT.MAX(14) ) MAX(14) = INTER
 5000     CONTINUE
C NOW MEAN OF THE ABSOLUTE PERFORMANCE DEVIATION
 5100     CONTINUE
          IF(SAMPLE(16).GT.0) GO TO 6000
C NO ABS PERF DEVS
          DO 5500 I = 1,ROSTER
          SECOND(7,I) = -888
 5500     CONTINUE
          GO TO 7100
 6000     CONTINUE
          GRADE = SUM(16) / SAMPLE(16)
          AVG  = IROUND(GRADE)
C NOW CALCULATE RELATIVE PERFORMANCE DEVIATION
          DO 7000 STDENT = 1,ROSTER
          IF(SECOND(6,STDENT).NE.-999) GO TO 6500
          SECOND(7,STDENT) = -999
          GO TO 7000
```

AIMS III SOURCE STATEMENT LISTING

```
  6500     CONTINUE
           GRADE = SECOND(6,STDENT) - AVG
           INTER = IROUND(GRADE)
           SECOND(7,STDENT) = INTER
           SAMPLE(17) = SAMPLE(17) + 1
           SUM(17) = SUM(17) + GRADE
           SUMSQ(17) = SUMSQ(17)+(GRADE * GRADE )
           IF(INTER.LT.MIN(17) ) MIN(17) = INTER
           IF(INTER.GT.MAX(17) ) MAX(17) = INTER
  7000     CONTINUE
C NOW CALCULTE MEANS, STD DEVS, ETC.
  7100     CONTINUE
           DO 8000 TEST = 11,17
           N = SAMPLE(TEST)
           IF(N.LE.0) GO TO 7500
           A = SUM(TEST)
           G = A/N
           MEAN(TEST) = IROUND(G)
           IF(N.GT.1) GO TO 7300
           SDEV(TEST) = 0
           GO TO 8000
  7300     CONTINUE
           B = A * A
           C = N * SUMSQ(TEST)
           D = N * (N-1)
           SS = ( C -B ) / D
           SD = SQRT(SS)
           SDEV(TEST) = IROUND(SD)
           GO TO 8000
  7500     CONTINUE
           MEAN(TEST) = -999
           SDEV(TEST) = -999
           MAX(TEST) = -999
           MIN(TEST) = -999
  8000     CONTINUE
C WE WILL NOW PLACE THE QUESTION COUNTERS ON DISK
           DO 10000 TEST = 1, DECKS
           NUM = TABLE(2,TEST)
           INK = TABLE(1,TEST)
           DO 10000 QST = 1, NUM
           IREC = QST + INK - 1
C READ A QUESTION RECORD
           READ(IQUEST'IREC) RECORD
           DO 8500 I = 1,11
           J = I + 29
           RECORD(J) = QSTION(I,QST,TEST)
  8500     CONTINUE
           DO 8800 I = 24,25
           J = I - 12
           T = QSTION(J+2,QST,TEST)
           IF(T.GT.0.99) GO TO 8600
           RECORD(I) = -999
           GO TO 8800
  8600     CONTINUE
           GRADE = QSTION(J,QST,TEST) / T
           RECORD(I) = IROUND(GRADE)
  8800     CONTINUE
           DO 9000 I = 14,17
           J = I + 12
```

```
          RECORD(J) = QSTION(I,QST,TEST)
 9000     CONTINUE
C PER CENT RIGHT
          INTER = RECORD(26) + RECORD(27)
          IF(INTER.GT.0) GO TO 9100
          RECORD(21) = -999
          GO TO 9200
 9100     CONTINUE
          S = RECORD(26)
          GRADE = ( (S / INTER) * 100.0 )
          RECORD(21) = IROUND(GRADE)
 9200     CONTINUE
          INTER = SDEV(TEST)
          IF(INTER.GT.0) GO TO 9300
          RECORD(22) = -999
          GO TO 9500
C VALIDITY
 9300     CONTINUE
          IF(RECORD(24).EQ.-999.OR.RECORD(25).EQ.-999) GO TO 9400
          S = RECORD(24) - RECORD(25)
          T = RECORD(26) + RECORD(27)
          T1 = RECORD(26) / T
          T2 = 1.0 - T1
          T = T1 * T2
          GRADE =  ((SQRT(T) / INTER) * 100.0) * S
          RECORD(22) = IROUND(GRADE)
          GO TO 9500
 9400     CONTINUE
          RECORD(22) = -999
 9500     CONTINUE
          RECORD(23) = SDEV(TEST)
          WRITE(IQUEST'IREC) RECORD
10000     CONTINUE
C GET THE COURSE
          READ(ISTDNT'1) RECORD
          IDENT(5) = RECORD(5)
          IDENT(1) = LESSON
          IDENT(2) = - 4
          IDENT(3) =  0
          IDENT(4) = 0
C NOW PLACE GRADES ON DISK
          NUMB = NSTUD + 5
          IREC =  (( LESSON - 1) * NUMB ) + 1
C WRITE SAMPLE SIZE
          WRITE(ISCORE'IREC) IDENT , SAMPLE
C WRITE MEAN
          IDENT(2)= -3
          IREC = IREC + 1
          WRITE(ISCORE'IREC) IDENT , MEAN
C WRITE MIN
          IDENT(2) = -2
          IREC = IREC + 1
          WRITE(ISCORE'IREC) IDENT , MIN
C WRITE MAX
          IDENT(2) = -1
          IREC = IREC + 1
          WRITE(ISCORE'IREC) IDENT , MAX
C WRITE SDEV
          IDENT(2) = 0
          IREC = IREC + 1
          WRITE(ISCORE'IREC) IDENT , SDEV
C NOW WRITE THE INDIVIDUAL SCORES AND RETURN
          DO 12000 STUENT = 1,ROSTER
          IDENT(2) = STUENT
          IREC = IREC + 1
          WRITE(ISCORE'IREC) IDENT , (ANSWER(I,STUENT),I=1,10 ) ,
     2    (SECOND(J,STUENT),J = 1,10)
12000     CONTINUE
          RETURN
          END
          FUNCTION IROUND(E)
          IF(E.EQ.0.0) GO TO 1000
          IROUND = E + (0.5 * ( E / ABS(E ) ) )
          RETURN
 1000     CONTINUE
          IROUND = 0
          RETURN
          END
/*
 PHASE AIMSLIST,AIMSHEAD
// EXEC FFORTRAN
          SUBROUTINE RLIST(LESSON)
C THIS ROUTINE PROVIDES THE USER WITH THE FACILITY TO LIST AN AIMS DATA
C TAPE. IT USES TAPSVC TO PROVIDE THE TAPE HANDLING AND LSTCDE
C TO FORMAT THE ANSWERS IN A USER READABLE FORMAT
C
          COMMON/FILES/IN(2),IOUT,IRESP,IN1,IHEAD,IN2(2),ISTUNT,IN3(2),
     2    ISYS,IN4(3)
          INTEGER LNAME(2),DATES(6)
          INTEGER * 2 PLESS , NSTUD
          INTEGER * 2 DATA(65,10) , NOS(50), PTRS(50)
          DATA LNAME/'LIST','  '/
          INTEGER SCRC1(2,2) , SLINE(2,10)  ,DECK , BLANK ,ANS(2,48,10)
          DATA SCR01/'SING','LE','DOUB','LE'/, BLANK/'  '/
          INTEGER FLAG ,STUENT , COUNT , SLASH , DOT
          DATA SLASH/'/'/ , DOT/'.'/
          INTEGER * 2  WASTE(5) , NAME(13)
C CHECK THE LESSON NUMBER
          READ(ISYS'2) PLESS
          IF(LESSON.GT.0.AND.LESSON.LE.PLESS) GO TO 250
          I = LESSON
          J = PLESS
          CALL ERROR(LNAME,0,1,I,J)
          RETURN
  250     CONTINUE
          READ(ISYS'3) PTRS
          READ(ISYS'4) NOS
          I1 = PTRS(LESSON)
          I2 = NOS(LESSON) + I1 - 1
          NOHEAD = NOS(LESSON)
          J = 0
          DO 500 I = I1,I2
          J = J + 1
          READ(IHEAD'I) (DATA(K,J),K=1,65)
  500     CONTINUE
C WE'VE READ THE HEADERS NOW PRINT THEM
          IPAGE = 1
          CALL INFO(DATES)
          WRITE(IOUT,501) DATES,IPAGE
  501     FORMAT(1H1,5X,'***  AIMS LESSON LISTING  ***',54X,'JOB  ',2A4,
     2    2X, 2A4 ,  1X , 2A4 / 90X ,'PAGE',I4 )
C WRITE OUT THE WHOLE THING
          WRITE(IOUT,95)  ( I, I = 1, NOHEAD)
          WRITE(IOUT,88)  (DATA(1,I) , I = 1, NOHEAD )
          WRITE(IOUT,89)
          WRITE(IOUT,91)  (DATA(3,I) , I = 1, NOHEAD )
          WRITE(IOUT,92)  (DATA(4,I) , I = 1, NOHEAD )
          WRITE(IOUT,93)  (DATA(5,I) , I = 1, NOHEAD )
          WRITE(IOUT,96)  (DATA(6,I) , I = 1, NOHEAD )
          DO 600 I = 1 , NOHEAD
          K = DATA(10,I) + 1
          DO 600 J = 1 , 2
          SLINE(J,I)  = SCROI(J,K)
  600     CONTINUE
          WRITE(IOUT,97) ((SLINE(J,I),J=1,2),I= 1, NOHEAD)
          WRITE(IOUT,99) (DATA(11,I),I = 1 , NOHEAD)
          WRITE(IOUT,119) (DATA(12,I), I= 1 ,NOHEAD)
          WRITE(IOUT,129) (DATA(13,I), I= 1 ,NOHEAD)
          DO  900  DECK = 1 , NOHEAD
          NOQ = DATA(13,DECK)
          DO 750 NN = 1 , NOQ
          JJ = NN + 17
          CALL LSTCDE(DATA(JJ,DECK),ANS(1,NN,DECK) )
  750     CONTINUE
          IF(NOQ.GE.48) GO TO  900
          NOQ = NOQ + 1
          DO 850 NN = NOQ,48
          DO 850 I = 1,2
          ANS(I,NN,DECK) = BLANK
  850     CONTINUE
  900     CONTINUE
          DO 1000 NN = 1 , 48
          IF(NN.NE.1) GO TO 950
          WRITE(IOUT,159)  NN , (  (ANS(I,NN,J),I=1,2 ), J= 1,NOHEAD)
          GO TO 1000
  950     CONTINUE
          WRITE(IOUT,169)  NN , (  (ANS(I,NN,J),I=1,2 ), J= 1, NOHEAD)
 1000     CONTINUE
C  NOW LIST THE TAPE
          CALL OPSYS('LOAD','AHEWTAPE')
          CALL TAPSVC(LESSON,FLAG)
          IF(FLAG.EQ.0) GO TO 1200
          CALL ERROR(LNAME,0,2)
          RETURN
 1200     CONTINUE
C GET THE NUMBER OF STUDENTS
          READ(ISYS'1) NSTUD
          IRCX = NOHEAD
          DO 8000 STDENT = 1, NSTUD
          NO = 0
 1500     CALL  TPDATA(STDENT, FLAG,LRI,COUNT,DATA)
          IF(FLAG.EQ.0) GO TO 2000
          IF(FLAG.EQ.1) GO TO 8000
          IF(FLAG.EQ.2) GO TO 9000
C TAPE ERROR
          CALL ERROR(LNAME,0,2)
          GO TO 9000
```

AIMS III SOURCE STATEMENT LISTING

```
C WE'VE GOT SOME DATA TO LIST
 2000    CONTINUE
         NO = NO + 1
         INDX = INDX + COUNT
         IPAGE = IPAGE + 1
         CALL INFO(DATES)
         WRITE(IOUT,501) DATES , IPAGE
         I1 =((NO - 1) * 10) +  1
         I2 = I1 + COUNT - 1
         WRITE(IOUT,95) (I , I = I1 , I2  )
         WRITE(IOUT,85) (DATA(1,I) , I = 1 ,COUNT )
         IF(NO.NE.1 ) GO TO 2250
         READ(ISFORT'STUDENT) XASTE , NAME
 2250    CONTINUE
         WRITE(IOUT,90 )  NAME, STUDENT
         WRITE(IOUT,91 )  (DATA(3,I) , I = 1 , COUNT)
         WRITE(IOUT,92 )  (DATA(4,I) , I = 1 , COUNT)
         WRITE(IOUT,93 )  (DATA(5,I) , I = 1 , COUNT)
         WRITE(IOUT,96 )  (DATA(6,I) , I = 1 , COUNT)
         WRITE(IOUT,98 ) ((DATA(I,J), I = 7,9 ), J =1,COUNT)
         WRITE(IOUT,109)  (DATA(12,I),I = 1 , COUNT )
         DO 2600 I = 1 , COUNT
         DO 2500 J = 13, 17
         IF(DATA(J,I).NE.0) GO TO 2500
         DATA(J,I) = - 999
 2500    CONTINUE
 2600    CONTINUE
         WRITE(IOUT,139)  (  DATA(14,I),SLASH, DATA(15,J),SLASH
     2   , DATA(13,I) , I = 1 , COUNT )
         WRITE(IOUT,149)  (  DATA(16,I) , DOT , DATA(17,I), I = 1,COUNT)
         DO 4000  NDK = 1 , COUNT
         NOQ = 24 * DATA(6,NDK)
         DO 3500 NN = 1, NOQ
         JJ = NN + 17
         CALL LSTCDE(DATA(JJ,NDK ),ANS(1,NN,NDK ) )
 3500    CONTINUE
         IF(NOQ.GE.48) GO TO 4500
         NOQ = NOQ + 1
         DO 3800 NN = NOQ,48
         DO 3700 I = 1,2
         ANS(I,NN,NDK ) = BLANK
 3700    CONTINUE
 3800    CONTINUE
 4000    CONTINUE
 4500    CONTINUE
         DO 5000 NN = 1 , 48
         IF(NN.NE.1) GO TO 4850
         WRITE(IOUT,159) NN , ( ( ANS(I,NN,J) , I =1,2), J = 1,COUNT  )
         GO TO 5000
 4850    CONTINUE
         WRITE(IOUT,169) NN, ( ( ANS(I,NN,J) ,I =1,2), J = 1 , COUNT)
 5000    CONTINUE
         IF(LRI. EQ. 0) GO TO 1500
 8000    CONTINUE
 9000    IPAGE = IPAGE + 1
         CALL INFO(DATES)
         WRITE(IOUT,501) DATES, IPAGE
         WRITE(IOUT,9001) IREX, NGHEAD
 9001    FORMAT(1X, 9(/) ,10X , 'THERE ARE', I4,
     2 'RECORDS '   ,          ', INCLUDING' , I4 , 1X , 'HEADERS'  )
  94    FORMAT(1X,'RECORD NUMBER',12X,10(3X,I2,4X) )
  95    FORMAT(1X,'DECK NUMBER',14X,10(3X,I2,4X) )
  88    FORMAT(1X,'LESSON',19X,10(3X,I2,4X)  )
  90    FORMAT(1X,'STUDENT NAME AND NUMBER',2X,15A2,2X,I3)
  91    FORMAT(1X,'SEGMENT',18X,10(3X,I2,4X) )
  92    FORMAT(1X,'TYPE',21X,    10(3X,I2,4X) )
  93    FORMAT(1X,'COURSE',19X,10(3X,I2,4X)  )
  96    FORMAT(1X,'NO. OF CARDS',13X,10(3X,I2,4X) )
  97    FORMAT(1X,'SINGLE OR DOUBLE',9X,10(1X,2A4) )
  98    FORMAT(1X,'IDENTIFICATION NUMBER',4X,10(2X,3A2,1X) )
  99    FORMAT(1X,'NO. OF SELECTIONS',8X,10(2X,I3,4X)  )
 109    FORMAT(1X,'NO. OF QUESTIONS ANSWERED',10(3X,I2,4X) )
 119    FORMAT(1X,'NO. OF QUESTIONS GRADED',2X,10(3X,I2,4X) )
 129    FORMAT(1X,'NO. OF QUESTIONS',9X,10(3X,I2,4X) )
 139    FORMAT(1X,'DATE MM/DD/YY',12X,10(1X,I2,A1,I2,A1,I2) )
 149    FORMAT(1X,'TIME HR.MN',15X,10(2X,I2,A1,I2) )
 159    FORMAT(1X,'QUESTION NUMBER',I3,10X,10(1X,2A4) )
 169    FORMAT(1X,15X, I3,10X, 10(1X,2A4) )
  89    FORMAT(1X,'HEADERS' )
        END
/*
// LBLTYP NSD(5)
// EXEC LNKEDT
/&
// JOB 304132        REPTMAIN
// ASSGN SYSLNK,X'192'
// DLBL IJSYSLN,'SYSLNK',60/365,SD
// EXTENT SYSLNK,000002,1,0,10,1000
// OPTION CATAL
   PHASE REPTAIMS,ROOT
// EXEC FFORTRAN
C          XXXXXXXXXX      OUTPUT  GENERATOR          XXXXXXXXXX
C      FILE 5  -  LESSON SCRATCH FILE
C      FILE 6  -  HEADER FILE
C      FILE 7  -  DIRECTORY FILE
C      FILE 8  -  QUESTION FILE
C      FILE 9  -  STUDENT BACKGROUND FILE
C      FILE 10 -  STUDENT SCORE FILE
C      FILE 11 -  TEXT FILE
C      FILE  12 - SYSTEM FILE
       DEFINE FILE 5(1500,33,U,I5)
       DEFINE FILE 6(600,33,U,I6)
       DEFINE FILE 7(200,100,U,I7)
       DEFINE FILE 8(4000,35,U,I8)
       DEFINE FILE 9(250,33,U,I9)
       DEFINE FILE 10(8040,33,U,I10)
       DEFINE FILE 11(400,23,U,I11)
       DEFINE FILE 12(100,25,U,I12)
       INTEGER*2 IRP(16)
C      THE FLOW OF THIS PROGRAM IS AS FOLLOWS
C
C           1.   INITIALIZE
C           2.   INPUT MONITOR DATA
C           3.   SELECT A REPORT
C           4.   INPUT REPORT DATA
C           5.   ORGANIZE THE DATA FOR OUTPUT
C           6.   PRODUCE A REPORT
C           7.   SELECT NEXT REPORT
```

AIMS III SOURCE STATEMENT LISTING

```
C READ REPORT MONITOR DATA FORMAT IS FOR USER REPORTS ONE THRU 16
      READ(1,97) (IRP(K),K=1,16)
C THE FOLLOWING DECISION BLOCK IS DESIGNED
C TO DO THE FOLLOWING
C
C          1.   TEST WHICH REPORT NUMBERS
C               ARE  REQUESTED
C          2.   DIRECT THE PROGRAM TO THE SECTION
C               WHICH WILL PRODUCE THE REQUESTED
C               REPORT
C
C          IRP1 - IRP16 REPRESENTS REPORTS
C          ONE THRU SIXTEEN RESPECTIVELY
C
      IRESPT=4
      DO 100 KIRP=1,16
      REWIND IRESPT
      N=IRP(KIRP)
      GO TO (70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85),N
      GO TO 100
C PRODUCE  REPORT  ONE
   70 CONTINUE
      CALL OPSYS('LOAD','REPT0001')
      CALL RSEP(IRP(KIRP))
      CALL REP1
      GO TO 100
C PRODUCE  REPORT  TWO
   71 CONTINUE
      CALL RSEP(IRP(KIRP))
      GO TO 100
C PRODUCE  REPORT  THREE
   72 CONTINUE
      CALL RSEP(IRP(KIRP))
      GO TO 100
C PRODUCE  REPORT  FOUR
   73 CONTINUE
      CALL OPSYS('LOAD','REPT0004')
      CALL RSEP(IRP(KIRP))
      CALL REP04
      GO TO 100
C PRODUCE  REPORT  FIVE
   74 CONTINUE
      CALL OPSYS('LOAD','REPT0004')
      CALL RSEP(IRP(KIRP))
      CALL REP05
      GO TO 100
C PRODUCE  REPORT  SIX
   75 CONTINUE
      CALL OPSYS('LOAD','REPT0004')
      CALL RSEP(IRP(KIRP))
      CALL REP06
      GO TO 100
C PRODUCE  REPORT SEVEN
   76 CONTINUE
      CALL OPSYS('LOAD','REPT0007')
      CALL RSEP(IRP(KIRP))
      CALL REP07
      GO TO 100
C PRODUCE  REPORT EIGHT
```

```
   77 CONTINUE
      CALL RSEP(IRP(KIRP))
      GO TO 100
C PRODUCE  REPORT NINE
   78 CONTINUE
      CALL RSEP(IRP(KIRP))
      GO TO 100
C PRODUCE  REPORT TEN
   79 CONTINUE
      CALL RSEP(IRP(KIRP))
      GO TO 100
C  PRODUCE  REPORT ELEVEN
   80 CONTINUE
      CALL CPSYS('LOAD','REPT0012')
      CALL RSEP(IRP(KIRP))
      CALL REP11
      GO TO 100
C PRODUCE  REPORT TWELVE
   81 CONTINUE
      CALL CPSYS('LOAD','REPT0012')
      CALL RSEP(IRP(KIRP))
      CALL REP12
      GO TO 100
C PRODUCE  REPORT THIRTEEN
   82 CONTINUE
      CALL RSEP(IRP(KIRP))
      GO TO 100
C PRODUCE  REPORT FOURTEEN
   83 CONTINUE
      CALL RSEP(IRP(KIRP))
      CALL CPSYS('LOAD','REPT0013')
      CALL REP14
      GO TO 100
C PRODUCE  REPORT FIFTEEN
   84 CONTINUE
      CALL CPSYS('LOAD','REPT0015')
      CALL RSEP(IRP(KIRP))
      CALL REP15
      GO TO 100
   85 CONTINUE
      CALL CPSYS('LOAD','REPT0015')
      CALL RSEP(IRP(KIRP))
      CALL REP16
  100 CONTINUE
   65 WRITE(3,96)
   96 FORMAT(/,1X,'END OF AIMS OUTPUT GENERATOR')
   97 FORMAT(10X,16I2)
      CALL EXIT
      END
      SUBROUTINE HEADPG(IRPT,IPGE)
      INTEGER*2 INFORM(12),DATE(4),TIME(4)
      INTEGER*2 ORP(7),IRP
     0INTEGER O(10)/'****',' AIM','S OU','TPUT',' GEN','ERAT','OR, ',
     1'REPO','RT N','O.  '/
      EQUIVALENCE (INFORM(5),DATE(1)),(INFORM(9),TIME(1))
      CALL INFO(INFORM)
      WRITE (3,10000) IRPT,TIME,DATE,IPGE
10000 FORMAT(1H1,/1H ,'     **** A.I.M.S. REPORT GENERATOR ****   REPORT
     2 NUMBER ',I2,' ****     TIME  - ',4A2,'     DATE  - ',4A2,'     PAGE
     3 NUMBER ',I4,//)
       IPGE=IPGE+1
       RETURN
       ENTRY RSLP(IRP)
       IOTPT=6
       DO 10 K=1,7
    10 URP(K)=IRP
       WRITE(IOTPT,20)
       DO 15 K=1,60
      WRITE(IOTPT,25) (URP(J),J=1,7),(Q(L),L=1,10),IRP,Q(1),
      1(URP(J),J=1,7),(Q(L),L=1,10),IRP,Q(1),(URP(J),J=1,7)
    15 CONTINUE
    20 FORMAT(1H1)
    25 FORMAT(1X,2(7I2,10A4,I2,A4),7I2)
       RETURN
       END
       FUNCTION SUBGRD(TP,H,A)
       INTEGER*2 H,A,TP
       INTEGER*2 IPASS,SUBGRD,ITEST,N,D(11)
       INTEGER*2 THW(4)/3,5,7,9/,SCLEHW(4)/100,80,60,20/
       INTEGER*2 SCLESG(4)/100,80,40,10/
       SUBGRD=0
       ITEST=IPASS(H,A,N,D)
       GO TO (10,10,30,40,10),TP
       WRITE(IOTPT,500)
   500 FORMAT(' *** ERROR *** INCORRECT TYPE SPEC, ')
       RETURN
    10 CONTINUE
     . IF(ITEST.EQ.1.AND.N.EQ.1) SUBGRD=100
       RETURN
    30 CONTINUE
       DO 32 K1=1,4
       IF(A.EQ. THW(K1)) SUBGRD=SCLEHW(K1)
    32 CONTINUE
       RETURN
    40 CONTINUE
       IF(ITEST.EQ.1) SUBGRD=SCLESG(N)
       RETURN
       END
       SUBROUTINE GETIT(REKORD,LESSON,SEGMNT ,TYPE,FILES,ERRORS)
       INTEGER*2 LESSON,SEGMNT,TYPE,ERRORS,QUESTN,  LPERM,CARD
       INTEGER*2 RECMBU(70),RECORD(65),REKORD(65),PTRS(50),NOS(50)
       INTEGER*2 WASTE(3)
       INTEGER SYSTEM,HEADER,QUESTO,TAPE
       INTEGER FILES(5)
       LCGICAL ONE
       SYSTEM=FILES(1)
       HEADER=FILES(2)
       QUESTO=FILES(3)
       TAPE=FILES(4)
       MIDDLE=1
       GO TO 1
       ENTRY GETMBU(RECMBU,LESSON,SEGMNT,TYPE,QUESTN,FILES,ERRORS)
       SYSTEM=FILES(1)
       HEADER=FILES(2)
       QUESTO=FILES(3)
       TAPE=FILES(4)
       MIDDLE=2
     1 READ (SYSTEM'1) LPERM,WASTE
```

```
 AIDS III SOURCE STATEMENT LISTING

        READ (SYSTEM'3) PTRS
        READ (SYSTEM'4) NOS
        IF ((LESSON.GT.0).AND.(LESSON.LE.LPGRM)) GO TO 2
        ERRORS=-1
        RETURN
      2 ISTART=PTRS(LESSON)
        IEND=ISTART+NOS(LESSON)-1
        DO 3 INDEX=ISTART,IEND
        READ (HEADER'INDEX) REKORD
        IF ((REKORD(3).EQ.SEGMNT ).AND.(REKORD(4).EQ.TYPE)) GO TO 4
      3 CONTINUE
        ERRORS=1
        RETURN
      4 ERRORS=0
        IF (VIDEOC.EQ.1) RETURN
        IPOINT=REKORD(6)+QUESTN-1
        IF (QUESTN.GT.REKORD(13)) GO TO 701
        READ(QUESTO'IPOINT) RECRBG
        RETURN
    701 ERRORS=-1
        RETURN
        ENTRY FINDIT(RECORD,LESSON,SEGMNT,TYPE,CARD,FILES,ERRORS,ONE)
        SYSTEM=FILES(1)
        HEADER=FILES(2)
        QUESTO=FILES(3)
        TAPE=FILES(4)
        ERRORS=0
        IF (.NOT.ONE) GO TO 100
        ONE=.FALSE.
        REWIND TAPE
    100 READ (TAPE,END=701) RECORD
        IF ((RECORD(1).EQ.LESSON).AND.(RECORD(3).EQ.SEGMNT).AND.(RECORD(4)
       2.EQ.TYPE)) GO TO 300
        IF((RECORD(1).GT.LESSON).AND.(CARD.EQ.2)) GO TO 701
        GO TO 100
    300 IF ((CARD.EQ.1).AND.(RECORD(2).EQ.0)) RETURN
        IF ((CARD.EQ.2).AND.(RECORD(2).NE.0)) RETURN
        IF (CARD.NE.1) GO TO 100
        BACKSPACE TAPE
        BACKSPACE TAPE
        GO TO 100
        END
        SUBROUTINE SUBMIT(TP,ST)
        IMPLICIT INTEGER*2 (K)
        LOGICAL*1 TEST(185,10)
        INTEGER*2 UID(12)/1,9,8,2,7,3,10,4,10,5,10,6/
        INTEGER*2 OUTCHK(10)
        INTEGER*2 TP(5),ST,SG
        INTEGER*2 CSN,RTP,RSG
        DO 10 K1=1,185
        DO 10 K2=1,10
        TEST(K1,K2)=.FALSE.
     10 CONTINUE
        RETURN
        ENTRY CHECK(CSN,RTP,RSG)
  C IF TEST( ) IS TRUE THE STUDENT SUBMITTED MATERIAL
        TEST(CSN,UID(RTP+2*(RSG-1)))=.TRUE.
        RETURN
        ENTRY GETCHK(CSN,OUTCHK)
       AIMS III SOURCE STATEMENT LISTING

      DO 20 K3=1,10
      OUTCHK(K3)=1
      IF(.NOT.TEST(CSK,K3)) OUTCHK(K3)=2
   20 CONTINUE
      OUTCHK(10)=1
      RETURN
      END
/*
// EXEC ASSEMBLY
IPASS    START
         CODE  ENTER
CORECT   EQU   4
STUDNT   EQU   6
ENTER    L     3,0(0,1)                 1ST PAR IS CORECT ANS
         LH    5,0(0,3)                 PLACE CORECT ANS IN GPR5
         L     3,4(0,1)                 2ND PAR IS STUDENT'S RESPONSE
         LH    7,0(0,3)                 PLACE STUDENT ANS IN GPR7
         NR    5,7                      CHECK CORRECT BITS
         L     11,12(0,1)               LOAD GPR11 WITH ADDR OF COUNTS
*                                       COUNTS ARE 4TH PARAMETER
         USING CNTAR,11                 TELL ASSEMBLER
         MVI   BLANK,X'00'              CLEAR COUNT AREA
         MVC   BLANK+1(21),BLANK
         SR    0,0                      CLEAR GPR0 FOR TOTAL CORRECT
         SR    3,3                      CLEAR GPR3 FOR TOTAL
         SLL   5,21                     LEFT SHIFT TO SET UP FOR COUNTS
         SLL   7,21
*                                       THIS LOOP BEGINS THE COUNTING
*                                       SET UP FOR THE LOOP
         LA    10,18(0,0)               THIS IS OUR INDEX GPR
         L     8,=F'-2'                 THIS IS THE INCREMENT
         L     9,=F'-1'                 THIS IS THE COMPARAND
LOOP     SR    CORECT,CORECT            CLEAR THE HI ORDER GPR'S
         SR    STUDNT,STUDNT
         SLDL  CORECT,1                 PLACE A BIT IN THE GPR
         SLDL  STUDNT,1                 ALSO FOR HIS RESPONSE
         AR    0,CORECT                 COUNT THE CORRECT BITS
         AR    3,STUDNT                 COUNT THE STUDENT RESPONSE
         STH   STUDNT,COUNTS(10)        AND COUNT THE RESPONSE
         BXH   10,8,LOOP                FINISH LOOP
         C     3,=F'0'
         BC    X'2',SOME                IF NONE OTHERS,THEN A BLANK
         L     4,=F'1'
         STH   4,BLANK                  COUNT THE BLANK
SOME     L     4,8(0,1)                 GET ADR OF 3RD PARAMETER
         STH   3,0(0,4)                 PLACE RESPONSE CNT IN 3RD PAR
         HOME
CNTAR    DSECT
BLANK    DS    1H
COUNTS   DS    10H
         END
/*
   INCLUDE RIGHT
   INCLUDE INFO
   PHASE REPT0001,*
// EXEC FFORTRAN
      SUBROUTINE REP1
      DIMENSION ALINE(32)
      DIMENSION ONE(3),TWO(3),THR(3),FOU(3),FIV(3)
   AIMS III SOURCE STATEMENT LISTING

      INTEGER FILES(5)
      INTEGER*2 LESSON,SEGMNT,TYPE,ERRORS,QUESTN
      INTEGER * 2 RP,LN,ST,SC,CSN,T1,T2,T3,T4,T5
      INTEGER*2 RIGHT,ITEST,GNO,COUNT
      INTEGER * 2 HOR(48),ANS(48),RECH(65),RECR(65),RECM(65)
      INTEGER * 2 KEY(17),PRE(15)
      INTEGER * 2 TP(5),CARD,KTP,KST
      INTEGER * 2 CRNM(6)
      INTEGER * 2 GRD
      INTEGER * 2 NAMENO(23)
      INTEGER * 2 SECTN(2),GROUP(2),BACK(65)
      DATA ONE/'PRE ','TEST','    '/,TWO/'POST',' TES','T   '/,
     1THR/'HOME','WORK','    '/,FOU/'STUD','Y GU','IDE '/,
     2FIV/'ASSI','GNME','NT  '/
      DATA ALINE/32*'----'/
      LOGICAL ONE
      EQUIVALENCE ( HOR(1), RECH(18)) , ( GNO , RECH(13))
      EQUIVALENCE ( KEY(1), RECM(41)) , ( PRE(1), RECM(58))
      EQUIVALENCE ( ANS(1), RECR(18)) , ( CSN , RECR(2)) , ( GRD , RECR(
     110))
      EQUIVALENCE (NAMENO(1) , BACK(1)) , (SECTN(1) , BACK(42)),
     1(GROUP(1) , BACK(44))
      READ(1,90) RP,LN,ST,SC,(TP(K),K=1,5),(CRNM(J),J=1,6)
   90 FORMAT(I2,7X,3I2,5I1,5X,6A2)
      LESSON = LN
      FILES(1) = 12
      FILES(2) =  6
      FILES(3) =  8
      FILES(4) =  4
      FILES(5) =  0
      GRAD = 0.00
      CARD = 2
      COUNT     =  0
      IRPT = 1
      IPGE = 1
      IOUT = 3
      ONE = .TRUE.
      DO 18 KTP=1,5
      IF(TP(KTP).LE.0.OR.TP(KTP).GT.5) GO TO 18
      TYPE= TP(KTP)
      SEGMNT=1
      IF(TP(KTP).NE.4) GO TO 17
      DO 16 KST=1,ST
      SEGMNT=KST
   17 CONTINUE
      CALL RSEP(RP)
      CALL GETIT (RECH,LESSON,SEGMNT,TYPE,FILES,ERRORS)
      IF(ERRORS.EQ.1) GO TO 12
      IF(ERRORS.EQ.-1)GO TO 12
    8 CALL FINDIT (RECR,LESSON,SEGMNT,TYPE,CARD,FILES,ERRORS,ONE)
C ERROR RETURN CHECK FOR FINDIT
      IF (ERRORS.EQ.-1) GO TO 10
C TITLES FOR REPORT ONE
      CALL HEADPG(IRPT,IPGE)
      INN = CSN
      READ (9'INN) (BACK(J),J=1,65)
      WRITE(3,100) (CRNM(J),J=1,6)
  100 FORMAT(40X,'STUDENT PERFORMANCE ANALYSIS FOR COURSE ',6A2)
      GO TO (26,27,28,29,30),TYPE
```

AIMS III SOURCE STATEMENT LISTING

```
  26 WRITE(IOUT,102) LN,(ONE(J),J=1,3),SEGMNT
     GO TO 31
  27 WRITE(IOUT,102) LN,(TWO(J),J=1,3),SEGMNT
     GO TO 31
  28 WRITE(IOUT,102) LN,(THR(J),J=1,3),SEGMNT
     GO TO 31
  29 WRITE(IOUT,102) LN,(FOU(J),J=1,3),SEGMNT
     GO TO 31
  30 WRITE(IOUT,102) LN,(FIV(J),J=1,3),SEGMNT
 102 FORMAT(1H0,59X,'VOLUME ',I2,12X,3A4,13X,'SEGMENT ',I2)
  31 CONTINUE
     WRITE(IOUT,117) (ALINE(J),J=1,32)
 117 FORMAT(1X,32A4)
     WRITE(IOUT,115)
 115 FORMAT(1H0,1X,'CSN',16X,'NAME',16X,'ID NUMBER',22X,'SECTION',25X,
    1'GROUP')
     WRITE(IOUT,116) CSN,(NAMENO(J),J=1,23),(SECTN(I),I=1,2),
    1(GROUP(K),K=1,2)
 116 FORMAT(1X,I4,23A2,22X,2A2,28X,2A2)
     WRITE(IOUT,117) (ALINE(J),J=1,32)
     WRITE(IOUT,103)
 103 FORMAT(1X,'QUESTION',7X,'ANSWER C/W',11X,'BEHAVIORAL OBJECTIVE',
    119X,'MESSAGE')
     DO 6 K = 1,QNO
     QUESTN = K
     ITEST = SUBGRD(TYPE,HOR(K),ANS(K))
     IF (ITEST.EQ.0) GO TO 7
     COUNT = COUNT+ ITEST
     WRITE(IOUT,107) K
 107 FORMAT(3X,I2,16X,'C')
     GO TO 6
   7 CALL GETMBO(RECN,LESSON,SEGMNT,TYPE,QUESTN,FILES,ERRORS)
     WRITE(IOUT,108) K,(KEY(J),J=1,17),(PRE(L),L=1,8)
 108 FORMAT(3X,I2,16X,'W',8X,17A2,6X,13A2)
   6 CONTINUE
     GRAD = COUNT/QNO
     IF(GRD.EQ.0) GO TO 21
     WRITE(IOUT,105) GRD
 105 FORMAT(1X,'YOUR GRADE IS  ',I3)
     WRITE(IOUT,104) GRAD
 104 FORMAT(1H0,1X,'YOUR RAW SCORE IS  ',F5.0)
     GO TO 22
  21 WRITE(IOUT,106) GRAD
 106 FORMAT(1H0,1X,'YOUR GRADE IS  ',F5.0)
  22 CONTINUE
     I  (COUNT/100+2).LT.QNO) WRITE(IOUT,999)
 999 FORMAT(1H0,1X,'YOU MUST SEE YOUR INSTRUCTOR THIS WEEK TO DISCUSS T
    1HIS TEST.')
     GRAD = 0.00
     COUNT = 0
     GO TO 8
C INSERT BEFORE RETURN
  12 WRITE(3,113) ERRORS
 113 FORMAT(1X,'GETIT ERROR LEVEL = ',I2)
     GO TO 11
  10 WRITE(3,112) ERRORS
 112 FORMAT(1H1,'FINDIT ERROR LEVEL = ',I2)
  98 FORMAT(1H1)
  11 CONTINUE
```

FILMED FROM BEST AVAILABLE COPY

```
          AIMS III SOURCE STATEMENT LISTING

        REWIND 4
        IF(TP(KTP).NE.4) GO TO 18
     18 CONTINUE
     16 CONTINUE
      / RETURN
        END
   /*
     PHASE REPT0004,REPT0001
   // EXEC FFORTRAN
        SUBROUTINE REP04
        INTEGER * 2   IPASS,NP,D(11)
        INTEGER * 2   MARK/'***'/,BLANK/'   '/
        INTEGER * 2   GRAD,KLN,CSN,KSC,KTP,EG,LN,ST,SN,CTQT,FP,RP
        INTEGER * 2   CELL(40),STAR(40,20),TP(5),NAME(6)
        INTEGER * 2   HDRREC(65),RESREC(65),HDR(48),ANS(48),
       1              ERRORS,RIGHT,ITEST,COUNT,CARD
        INTEGER       FILES(5)/12,6,6,4,0/
        INTEGER * 2   QNC,CLWTH,CLWTH2,CELNUM(20),ZERO,MAXVAL,SCALE
        INTEGER * 2   INITL,ICNTL(20),IACK
        INTEGER  *  2 CHECK
        INTEGER  *  2     SCORES(40)
        INTEGER * 2  SCORE,SUBGRD
       CINTEGER     TYPNAM(3,5)/'PRE-','TEST','    ',
       1                          'POST',' TES','T   ',
       2                          'HOME',' WOR','K   ',
       3                          'STUD','Y GU','IDE ',
       4                          'ASSI','GNME','NT  ' /
        LOGICAL ONE
        EQUIVALENCE (HDR(1),HDRREC(18)),
       1            (ANS(1),RESREC(18)),
       2            (CSN   ,RESREC( 2)),
       3            (QNC   ,HDRREC(13))
        INPT  =  1
        IOUT  =  3
        CHECK  =  0
        CARD = 2
        IZERO = 0
        IRPT=4
        IPGE=1
        SCORE = 0.00
        ONE = .TRUE.
        CELL(1) = 0
        DO 98 K15=1,40
        DO 97 K14 = 1,20
        CELNUM(K14)=0
        SCORES(K15) = 0
        STAR(K15,K14) = 0
     97 CONTINUE
     98 CONTINUE
  C
  C READ IN REPORT CONTENT CARD
  C      REQUIRED INFORMATION -
  C                1 REPORT NUMBER    -  RP
  C                2 LESSON NUMBER    -  LN
  C                3 SEGMENT          -  ST
  C                4 TYPE             -  TP
  C                5 COURSE NAME      -  NAME
  C
        READ(INPT,90) RP,LN,ST,SN,(TP(J),J=1,5),(NAME(I),I=1,6)
      AIMS III SOURCE STATEMENT LISTING

   90 FORMAT(I2,7X,3I2,5I1,9X,6A2)
      DO 95 JJ = 1,5
      IF(TP(JJ).EQ.0) GO TO 95
C
C USE GETIT TO OBTAIN HEADER RECORD OF REQUIRED TEST
C
      CALL GETIT(HDRREC,LN,ST,TP,FILES,ERRORS)
      IF(ERRORS.EQ.1) GO TO 200
      IF(ERRORS.EQ.-1) GO TO 201
C
C COMPUTE CELL WIDTH AND CELL VALUES
C          SET UPPER CELL VALUE TO 100
C                THERE IS ONE CELL FOR EACH QUESTION IN THE TEST
C
      CLWTH = 100 / QNO
      DO 30 K4 = 1,QNO
      KK4 = 1
      IF(K4.GT.1)  KK4 = K4  - 1
      CELL(K4) = CELL(KK4) + CLWTH
   30 CONTINUE
      CELL(QNO) = 100
C
C READ RESPONSE RECORD
C         GRADE STUDENT
C                ASSIGN GRADE TO CELL
C                      REPEAT UNTIL ALL STUDENTS ARE PROCESSED
C
  220 CALL FINDIT(RESREC,LN,ST,TP,CARD,FILES,ERRORS,QNO)
      IF(ERRORS.EQ.-1) GO TO 11
      COUNT = 0
      DO 210 K = 1,QNO
      ITEST = SUBGRD(TP,HDR(K),ANS(K))
      IF(ITEST.EQ.0) GO TO 210
      COUNT = COUNT+ ITEST
  210 CONTINUE
      SCORE = COUNT/QNO
      DO 40 K1 = 1,QNO
      IF(SCORE.NE.0) GO TO 42
      IZERO = IZERO + 1
      K1 = QNO
      GO TO 40
   42 IF(SCORE.GT.CELL(K1)) GO TO 40
      CELNUM(K1) = CELNUM(K1) + 1
      K1 = QNO
   40 CONTINUE
      GO TO 220
C
C SCALE CHART ACCORDING TO VALUES OF CELNUM
C         PRESENT MAXIMUM VALUES ARE - 40,80,120,160
C
   11 DO 61 K5 = 1,QNO
      IF (CELNUM(K5).GT.40) GO TO 60
   61 CONTINUE
      MAXVAL = 40
      SCALE  = 1
      GO TO 62
   60 DO 63 K6 = 1,QNO
      IF (CELNUM(K6).GT.80) GO TO 64
   63 CONTINUE
```

AIMS III SOURCE STATEMENT LISTING

```
      MAXVAL = 80
      SCALE  = 2
      GO TO 62
   64 DO 65 K7 = 1,QNO
      IF (CELNUM(K7).GT.120) GO TO 66
   65 CONTINUE
      MAXVAL = 120
      SCALE  = 3
      GO TO 62
   66 DO 67 K8 = 1,QNO
      IF (CELNUM(K8).GT.160) GO TO 99
   67 CONTINUE
      MAXVAL = 160
      SCALE  = 4
      GO TO 62
   99 WRITE(IOUT,100)
      WRITE(IOUT,107) (CELNUM(K11),K11=1,QNO)
  107 FORMAT(1H0,20I5)
  100 FORMAT(1H1,'ERROR , MAXVAL EXCEEDS LIMIT')
      RETURN
   62 CONTINUE
      DO 41 L1 = 1,40
      SCORES(L1) = (MAXVAL) - (L1 * SCALE) +  SCALE
   41 CONTINUE
      DO 70 K10 = 1,QNO
      DO 72 L2 = 1,40
      IF(CELNUM(K10).GE.SCORES(L2))  STAR(L2,K10) = MARK
   72 CONTINUE
   70 CONTINUE
C
C PRINT TITLES , MATRIX , STATISTICS
C
      CALL HEADPG(IRPT,IPGE)
      WRITE(IOUT,101) (NAME(I),I=1,6)
  101 FORMAT(1H0,28X,'AIMS HISTOGRAM ANALYSIS FOR ',6A2)
      WRITE(IOUT,92) LN,(TYPNAM(JJJ,TP(JJ)),JJJ=1,3),ST
   92 FORMAT(1H0,20X,'VOLUME, ',I3,15X,'TYPE, ',3A4,15X,'SEGMENT, ',I3)
      WRITE(IOUT,300)
  300 FORMAT(20X ,72('-'))
      DO 80  K11 = 1,40
      WRITE (IOUT,103) SCORES(K11),(STAR(K11,K12),K12=1,QNO)
  103 FORMAT(I3,12X,20(A1,3X))
   80 CONTINUE
      WRITE(IOUT,104)
  104 FORMAT(2X,'0',71('-'))
      WRITE(IOUT,105)  (CELL(K13),K13= 1,QNO)
  105 FORMAT(' SCORES',6X,20(I3,1X))
      WRITE(IOUT,106) (CELNUM(K13),K13 = 1,QNO)
  106 FORMAT(' STUDENTS',4X,20(I3,1X))
      WRITE(IOUT,109) IZERO
  109 FORMAT(1H0,' NUMBER OF ZERO VALUES IN SAMPLE = ',I3)
   95 CONTINUE
      GO TO 96
  200 WRITE(IOUT,230)
  230 FORMAT(1H1,'GETIT = 1')
      GO TO 96
  201 WRITE(IOUT,231)
  231 FORMAT(1H1,'GETIT = -1')
      GO TO 96
```

```
        AIMS III SOURCE STATEMENT LISTING

  202 WRITE(IOUT,232)
  232 FORMAT(1H1,'FINDIT = -1')
   95 RETURN
      END
      SUBROUTINE RLP05
C          COURSE STRUCTURE SUMMARY
      INTEGER*2 LESSON,SEGMNT,TYPE,CRNO,QUESTN,TO(2)
      INTEGER*2 DESC(17),PRESC(13),CRNM(6),RECM(70)
      INTEGER*2 CLES
      INTEGER * 2 ANSWER
      INTEGER * 2 A/'A'/,B/'B'/,C/'C'/,D/'D'/,E/'E'/
      EQUIVALENCE (LESSON , RECM(1))  ,(SEGMNT , RECM(2)),
     1(TYPE , RECM(3)) , (CRNO , RECM(4)) , (QUESTN , RECM(5)),
     2(TO(1), RECM(6)) , (DESC(1), RECM(41)), (PRESC(1), RECM(58))
     3,(ANSWER , RECM(12))
      IPGE=1
      IRPT=5
      IQUEST = 8
      IRPT   = 5
      IPGE   = 1
      IOUT   = 3
      INPT   = 1
      CRNO   = 0
      ILES   = 1
      CLES   = 0
      CNTR = 0
      READ(INPT,105) (CRNM(J),J=1,6)
  105 FORMAT(29X,6A2)
      CALL HEADPG(IRPT,IPGE)
   12 WRITE(IOUT,100)
  100 FORMAT(     40X,'COURSE STRUCTURE SUMMARY')
      READ(IQUEST'1) (RECM(J),J=1,70)
      WRITE(IOUT,101) CRNO,(CRNM(J),J=1,6)
  101 FORMAT(1H0,27X,'COURSE NO. ',I2,24X,'COURSE NAME  ',6A2)
   18 WRITE (IOUT,102)
  102 FORMAT(1H0,'VOLUME SEGMENT TYPE QUESTION  TO GO     DESCRIPTION
     1                    PRESCRIPTION',16X,'ANSWER')
   14 READ(IQUEST'ILES) (RECM(J),J=1,70)
      ILES = ILES + 1
      IF(ANSWER.EQ.3) ANSWER = A
      IF(ANSWER.EQ.5) ANSWER = B
      IF(ANSWER.EQ.9) ANSWER = C
      IF(ANSWER.EQ.17)ANSWER = D
      IF(ANSWER.EQ.33)ANSWER = E
      IF(LESSON.EQ.0) GO TO 20
      WRITE(IOUT,103) LESSON,SEGMNT ,TYPE,QUESTN,(TO(J),J=1,2),
     1(DESC(J),J=1,17),(PRESC(J),J=1,13)
     2,ANSWER
  103 FORMAT(2X,I2,6X,I2,4X,I2,5X,I2,5X,2I3,3X,17A2,13A2,4X,A4)
      CNTR = CNTR + 1
      IF(CNTR.EQ.45) GO TO 16
      GO TO 14
   16 CNTR = 0
      CALL HEADPG(IRPT,IPGE)
      GO TO 18
   20 WRITE(IOUT,104)
  104 FORMAT(1H0,40X,'END OF PROCE
      RETURN
      END
      AIMS III SOURCE SYSTEM LISTING

      SUBROUTINE REPO6
C     ***  ITEM ANALYSIS FOR AIMS VERSION 3  ***
      INTEGER FILES(5)
      INTEGER RTAPE
      INTEGER * 2  RESPSE(65),ANSWER(48),LESSON,SEGMNT,TYPE,TYPES(5)
      INTEGER*2 QTR(48,10),TOTQTR(48),TOTRES(10),REPNO,SECTN,CRSNAM(6)
      INTEGER*2 STAR(48,10),ERRORS,CARD,MARK,BLANK,ONE,IPASS,NOPUN
      INTEGER*2 HORREC(65),HOR(48),NUMSEL,DUMRES,HEAD,ANSPOS(11)
      INTEGER*2 ALPBET(10)/'A','B','C','D','E','F','G','H','I','J'/
      INTEGER*2 DUMMY/0/,INDANS,CNTMUL,CNTTST,TOTAL
      INTEGER*2 TOTSEL,INDRES,NUMZER
      INTEGER   DASH(10)/10*'----'/
     DINTEGER    TYPNAM(3,5)/'PRE-','TEST','    ',
     1                          'POST',' TES','T   ',
     2                          'HOME',' WOR','K   ',
     3                          'STUD','Y GU','IDE ',
     4                          'ASSI','GNME','NT  '/
      EQUIVALENCE (TOTSEL,HORREC(11))
      EQUIVALENCE ( ANSWER(1) , RESPSE(18))
      EQUIVALENCE(HOR(1),HORREC(18)) , (ONE , HORREC(13))
      DATA MARK/'***'/,BLANK/' '/
      LOGICAL ONE
      LNPT    = 6
      IPGE    = 1
      IOUT    = 3
      INPT    = 1
      RTAPE   = 4
      CARD = 2
      FILES(1)  =   12
      FILES(2)  =    6
      FILES(3)  =    8
      FILES(4)  =    4
      FILES(5)  =    0
      ONE = .TRUE.
C     ***   INITIALIZE  MATRICES   ***
      DO 5 I=1,48
      TOTQTR(I) =0
      DO 6 J=1,10
      TOTRES(J) =0
      QTR(I,J)  =  0
      STAR(I,J) =  BLANK
    6 CONTINUE
    5 CONTINUE
      TOTAL = 0
      NUMZER = 0
      CNTMUL=0
      CNTTST=0
C     ***   READ  IN  REPORT  CONTENT  CARD    ***
      READ(INPT,90) REPNO,LESSON,SEGMNT,SECTN,(TYPES(K),K=1,5),
     1(CRSNAM(J),J=1,6)
   90 FORMAT(I2,7X,3I2,5I1,9X,6A2)
C     ***   MAJOR  LOOP  IS  FOR  TYPE  NUMBER    ***
      DO 34 K=1,5
      TYPE = TYPES(K)
      IF(TYPES(K).EQ.0) GO TO 30
C     ***
C     ***   INSERT  ASTERISK  FOR  CORRECT  ANSWER    ***
      CALL GETIT(HORREC,LESSON,SEGMNT,TYPE,FILES,ERRORS)
      NUMSEL = TOTSEL/ONE
```

AIMS III SOURCE STATEMENT LISTING

```
      DO 36 J2= 1,GNO
      HEAD= HCN(J2)
      DUMRES = IPASS(DUMMY,HEAD,NUMPUN,ANSPUN)
      DO 110 J3 = 1,NUMSEL
      IF (ANSPUN(J3+1).EQ.1) STAR(J2,J3) = MARK
  110 CONTINUE
   36 CONTINUE
C    ***
C    ***   BEGIN  OUTPUT  SECTION    ***
      CALL HEADPG(INPT,IPGE)
      WRITE(IOUT,91)
   91 FORMAT(     50X,'AIMS ITEM ANALYSIS')
      WRITE(IOUT,92) LESSON,(TYPNAM(N,TYPE),N=1,3),SEGMNT
   92 FORMAT(1H0,20X,'VOLUME, ',I3,15X,'TYPE, ',3A4,15X,'SEGMENT, ',I3)
      WRITE (IOUT,190) NUMSEL
  190 FORMAT(35X,'NUMBER OF SELECTIONS PER QUESTION, ',I2)
      WRITE(IOUT,93)
   93 FORMAT(1H0,35X,'NUMBER OF TIMES EACH ANSWER CHOSEN')
      WRITE(IOUT,94)
   94 FORMAT(14X,'QUEST',30X,'ANSWERS')
      WRITE(IOUT,95) ALPBET
   95 FORMAT(14X,'NUMBER',3X,10(A1,5X),    'TOTAL ANSWERS')
C    ***    SET  UP  FOR  TAPE  READ   ***
C    ***
C    ***   READ  RESPONSE  TAPE  ***
   14 CALL FINDIT (RESPSE,LESSON,SEGMNT,TYPE,CARDS,FILES,ERRORS,GNO)
      IF (ERRORS.EQ.-1)GO TO 32
C    ***   FILL  IN  RESPONSE  MATRIX   ***
      CNTTST = CNTTST + 1
      DO 16 I = 1,GNO
      INDRES = ANSWER(I)
      DUMRES = IPASS(DUMMY,INDRES,NUMPUN,ANSPUN)
      IF (NUMPUN.GT.1) CNTMUL = CNTMUL + 1
      IF(NUMPUN.EQ.0) NUMZER = NUMZER + 1
      DO 120 J4 = 1,NUMSEL
      QTN(I,J4) = QTN(I,J4) + ANSPUN(J4+1)
  120 CONTINUE
   16 CONTINUE
      GO TO 14
C    ***
C    ***   COMPUTE  LINE  TOTALS   ***
   32 DO 18 IQ =1,GNO
      DO 20 JS =1,NUMSEL
      TOTQTN(IQ) = TOTQTN(IQ) + QTN(IQ,JS)
      TOTRES(JS) = TOTRES(JS) + QTN(IQ,JS)
   20 CONTINUE
   18 CONTINUE
      DO 200 J4=1,NUMSEL
  200 TOTAL = TOTAL + TOTRES(J4)
      DO 22 IQ=1,GNO
      WRITE (IOUT,150) IQ, (QTN(IQ,J),STAR(IQ,J),J=1,10),TOTQTN(IQ)
  150 FORMAT(1H0,16X,I2,10(3X,I3,A1),7X,I3)
   22 CONTINUE
C    ***
      WRITE(IOUT,160) DASH
  160 FORMAT(10X,'TOTAL',7X,10(A3,4X))
      WRITE(IOUT,170) TOTRES,TOTAL
  170 FORMAT(10X,'SELECTIONS ',10(I4,3X),3X,I5)
      WRITE(IOUT,220) CNTTST
      [illegible] THE SOURCE STATEMENT LISTING

  220 FORMAT(//,10X,'NUMBER OF WORDS PROCESSED, ',I4)
      WRITE(IOUT,180) CNTMUL
  180 FORMAT( /,10X,'NUMBER OF MULTIPLE SELECTIONS, ',I3)
      WRITE(IOUT,210) NUMZER
  210 FORMAT(/,10X,'NUMBER OF BLANK SELECTIONS, ',I3)
   34 CONTINUE
   26 REWIND NTAPE
   30 RETURN
      END
/*
   PHASE REPT0007,REPT0001
// EXEC FORTRAN
      SUBROUTINE REP07
      COMMON /FILES/IFILES(15)
      DIMENSION ITOS(200),IDATA(5),LEVELS(200)
      INTEGER * 2 NF,LA,CARN(6)
      READ(1,90) NF,LA,(CARN(J),J=1,6)
   90 FORMAT(I2,7X,I2,16X,6A2)
      DO 10 I=1,15
      IFILES(I)=I
   10 CONTINUE
      INLES = LA
      CALL DECIDE(INLES,ITOS,IDATA,LEVELS)
      IPT=IFILES(3)
      CALL DIVIDE(IDATA,ITOS,INLES,LEVELS)
      RETURN
      END
      SUBROUTINE DECIDE( INLES,ITOS,DATA,LEVELS )
      INTEGER HEADER,SYSTEM
      INTEGER LEVELS(200),ITOS(200),DATA(5)
      INTEGER TO,PAGE
      INTEGER*2 TOTBLE(48,10),HDRS(66,10),OCTBLE(10,2)
      INTEGER*2 TOCNTS(200),BCKG(65)
      INTEGER*2 NOS(40),PTRS(40),STUREC(261),RECS(65),RECH(65),RECQ(65)
      INTEGER*2 INFORM(12),DATE(4),TIME(4)
      INTEGER*2 ROSTER,LPERM,MPERM,LTEMP,KTEMP,ERRFLG,RIGHT
      INTEGER*2 ISTU1(65),ISTU2(65),ISTU3(65),ISTU4(65)
      EQUIVALENCE (INFORM(5),DATE(1)),(INFORM(9),TIME(1))
      EQUIVALENCE (ISTU1(1),STUREC(1)),(ISTU2(1),STUREC(66))
      EQUIVALENCE (ISTU3(1),STUREC(131)),(ISTU4(1),STUREC(196))
      COMMON /FILES/ ICD,IPC,IPT,IW1,ISCR,HEADER,IOCT,IQUES,IBKGD,ISCORE
     2,ITEXT,SYSTEM
      IWRONG=0
      TO=6
     *ITAPE=IW1
      IHEAD=HEADER
      ISYS=SYSTEM
       DATA(3)=0000
      IRPT=13
      PAGE=1
      DO 1 N=1,200
      TOCNTS(N)=0000
      ITOS(N)=0000
      LEVELS(N)=0000
    1 CONTINUE
      CALL HEADPG(IRPT,PAGE)
C
C     THE NEXT THING TO DO IS READ IN THE THRESHOLD LEVELS,
C   AND LIST THEM FOR THE USER'S USE
```

AIMS III SOURCE STATEMENT LISTING

```
C
      WRITE (IPT,14)
   14 FORMAT(1H ,40X,'TERMINAL OBJECTIVE THRESHOLD LEVEL SUMMARY',/40X,
     2'-------------------------------------------',/1H ,30X,'TERMINAL O
     3BJECTIVE',23X,'THRESHOLD LEVEL',/30X,'------------------',23X,'----
     4-----------',/1H0)
C
C          THAT HEADS THE PAGE
C
   15 READ (ICD,16,END=100) INDEX,ILEV
   16 FORMAT (2I3)
      IF (LEVELS(INDEX).GT.0) GO TO 7002
      LEVELS(INDEX)=ILEV
      GO TO 15
 7002 WRITE(IPT,7003) INDEX,ILEV,LEVELS(INDEX)
 7003 FORMAT (1H1,'**** ERROR ****     DUPLICATE THRESHOLD SPECIFICATION-
     2  T.O. = ',I3,' ATTEMPT D VALUE = ',I3,' ORIGINAL VALUE = ',I3,///
     3///////)
      CALL EXIT
  100 N=0
      DO 102 IN=1,200
      IF(LEVELS(IN).EQ.0000) GO TO 102
      N=N+1
      WRITE (IPT,101) IN,LEVELS(IN)
  101 FORMAT(1H ,37X,I3,35X,I3)
      IF(N.LE.47) GO TO 102
      CALL HEADPG(IRPT,PAGE)
      N=0
      WRITE(IPT,14)
  102 CONTINUE
C            SO MUCH FOR GETTING AND LISTING LEVELS
      READ (ISYS'1) ROSTER
      DATA ISTU1/65*0/
      IFINIS=ROSTER*2
      DO 3 ICLR1=1,IFINIS
      WRITE(ISCH'ICLR1)ISTU1
    3 CONTINUE
      READ (ISYS'2) LPERM,NPERM,LTEMP,NTEMP,ERRFLG
      READ (ISYS'3) PTRS
      READ (ISYS'4) NOS
      ISTART=PTRS(INLES)
      IEND=ISTART+NOS(INLES)-1
      IDIFF=ISTART-1
      IF ((INLES.GT.LPERM).OR.(INLES.LE.0)) CALL EXIT
   61 DO 2 INP=ISTART,IEND
      INDEX=INP-IDIFF
      READ (IHEAD'INP) (HDRS(N,INDEX),N=1,66)
      OCTBLE(INDEX,1)=HDRS(4,INDEX)
      OCTBLE(INDEX,2)=HDRS(5,INDEX)
      NQUES=HDRS(13,INDEX)
      IPOINT=HDRS(8,INDEX)
      IANP=IPOINT+NQUES-1
      DO 2 J=IPOINT,IANP
      READ (IQUES'J) RECQ
      INDX1=J-IPOINT+1
      TOTBLE(INDX1,INDEX)=RECQ(TO)
      ITO=RECQ(TO)
      TOCNTS(ITO)=TOCNTS(ITO)+1
    2 CONTINUE
```

```
      IPS III SOURCE STATEMENT LISTING

      REWIND ITAPE
   11 READ (ITAPE,END=7000) RECS
      IF (RECS(1).NE.INLES) GO TO 12
      IF (RECS(2).EQ.0) GO TO 11
      GO TO 13
   12 IF (RECS(1).GT.INLES) GO TO 7000
      GO TO 11
 7000 WRITE (IPT,7001) INLES
 7001 FORMAT (1H1,'NO RECORD OF LESSON ',I5,//////)
      CALL EXIT
   13 CONTINUE
C
C                 AT THIS POINT, I'VE GOT   1 THE FIRST KID
C                       TOTBLE  -           2 A TABLE OF TO'S VS. QUESTIONS
C                       DCTBLE               3 A TABLE OF DECKS VS. TYPES
C                       TOCNTS               4 T
C                       TOCNTS               4 A TABLE OF NO. OF QUEST'S /TO
C                       HDRS                 5 ALL THE HEADERS FOR A LESSON
C   NOW ITS TIME TO DO THE REAL WORK
  105 DO 106 IN=1,201
  106 STUREC(IN)=0000
  104 ITYPE=RECS(4)
      ISEQ=RECS(3)
      DO 107 IN=1,10
      IF(DCTBLE(IN,1).NE.ITYPE) GO TO 107
      IF(DCTBLE(IN,2).EQ.ISEQ) GO TO 108
  107 CONTINUE
      WRITE (IPT,7004) ITYPE,DCTBLE
 7004 FORMAT(1H1,'ILLEGAL TYPE NUMBER- ',I2,/1H0,'KNOWN TYPES ARE -',
     210I4,/////)
      CALL EXIT
  108 IDECK=IN
      NOQUES=HDRS(13,IDECK)
      DO 110 IN=1,NOQUES
      IF (HDRS(IN+17,IDECK).EQ.1) GO TO 110
      IXV=IN+17
      IF (RIGHT(RECS(IXV),HDRS(IXV,IDECK)).NE.IWRONG) GO TO 110
      ITONG=TOTBLE(IN,IDECK)
      STUREC(ITONG)=STUREC(ITONG)+1
  110 CONTINUE
      ISAVE=RECS(2)
      READ(ITAPE,END=8000) RECS
      IF (RECS(2).EQ.ISAVE) GO TO 104
C     IF IT GETS HERE, I'M DONE WITH THE KID
      DO 111 IN=1,200
      ITOP=STUREC(IN)
      IBOT=TOCNTS(IN)
      IF(IBOT .EQ. 0) GO TO 111
      STUREC(IN)= (100*ITOP)/IBOT
      IF (STUREC(IN).LE.LEVELS(IN)) GO TO 111
      STUREC(201)=1
      ITOS(IN)=ITOS(IN)+1
  111 CONTINUE
      IF(STUREC(201).GT.0) DATA(3)=DATA(3)+1
      IF(STUREC(201).EQ.0) GO TO 120
      CALL HEADPG(IRPT,PAGE)
      READ(IBKGD'ISAVE)BCKG
      WRITE (IPT,112) (BCKG(IRT),IRT=6,18),(BCKG(IRT),IRT=19,23),INLES
  112 FORMAT  (1H0,13A2,' I.D. NUMBER  ',5A2,/1H ,'THE STUDENT IDENTIFI
     2O ABOVE HAS PERFORMED BELOW PRE-SET LEVELS OF PERFORMANCE IN TERMI
     3NAL OBJECTIVE(S) IN LESSON ',I2,/1H0,'-----------------------------
     4-----------  PERFORMANCE DATA -------------------------------------
     5---',/1H0,5X,'T.O.',20X,'PERCENT ERR',20X,'CUTOFF LEVEL',/1H0)
      DO 113 IRT=1,200
      IF(STUREC(IRT).LE. 0000) GO TO 113
      WRITE (IPT,114)IRT,STUREC(IRT),LEVELS(IRT)
  114 FORMAT (1H ,5X,I3,25X,I3,30X,I3)
  113 CONTINUE
  120 IPOINT=(ISAVE*4)-3
      INEXT=IPOINT+1
      INEX=INEXT+1
      INX=INEX+1
      WRITE (ISCH'IPOINT) ISTU1
      WRITE (ISCH'INEXT) ISTU2
      WRITE(ISCH'INEX)ISTU3
      WRITE(ISCH'INX)ISTU4
      IF (RECS(1).GE.INLES) GO TO 8000
      GO TO 125
 8000  DATA(2)=ROSTER
       DATA(1)=0
      DO 8001 IRT=1,200
      IF (ITOS(IRT).EQ.0) GO TO 8001
      DATA(1)=DATA(1)+1
 8001 CONTINUE
      DO 8100 IND=1,INT
      IF(TOCNTS(IND).LE. 0000) GO TO 8100
      IF(ITOS(IND) .LE. 0000) GO TO 8100
      CALL READPG(IRPT,PAGE)
      WRITE(IPT,8002) INLES,IND,TOCNTS(IND),ITOS(IND),LEVELS(IND)
 8002 FORMAT (1H0,40X,'TERMINAL OBJECTIVE REMEDIAL SUMMARY',/1H0,20X,'LE
     2SSON ',I2,5X,'TERMINAL OBJECTIVE ',I3,4X,'COMPOSED OF ',I2,' QUEST
     3IONS.'//10X,'A TOTAL OF ',I3,' STUDENTS PERFORMED BELOW THE CUTOFF
     4 LEVEL SET AT ',I3//1X,'THE FOLLOWING STUDENTS HAVE PERFORMED BELO
     5W THE CUTOFF LEVEL-'//11X,'NAME OF STUDENT',12X,'I.D. NO.',3X,'PER
     6CENT ERROR'/7X,26('-'),2X,10('-'),5X,'----------'//)
      DO 8005 IBMT=1,ROSTER
      IPOINT=(IBMT*4)-3
      INEXT=IPOINT+1
      INEX=INEXT+1
      INX=INEX+1
      READ(ISCH'IPOINT)ISTU1
      READ(ISCH'INEXT)ISTU2
      READ(ISCH'INEX) ISTU3
      READ(ISCH'INX)ISTU4
      IF(STUREC(IND).LE. LEVELS(IND)) GO TO 8005
      READ (IBKGD'IBMT)BCKG
      WRITE(IPT,8006) (BCKG(INK),INK=6,23) , STUREC(IND)
 8006 FORMAT(1H0,6X,13A2,2X,5A2,5X,I3)
 8005 CONTINUE
 8100 CONTINUE
      RETURN
      END
      SUBROUTINE DIVIDE (IDATA,ITOS,INLES,LEVELS)
C
C THIS PROGRAM WILL ASSIGN THE TO'S TO BE TAUGHT IN THE RESPECTIVE
C    REMEDIAL SESSIONS. IT CAN HANDLE UP TO 10 SESSIONS AND 99 TO'S.
C
      DIMENSION ICNT(10),IVEX(200,2),LEC(10,20),ICLASS(10,20),IDATA(5)
      AIRS III SOURCE STATEMENT LISTING

      DIMENSION ITCS(200),LEVELS(200),LIC(20),ITEMP(20)
      INTEGER PAGE
      INTEGER*2 INFORM(12),DATE(4),TIME(4),RECD(65)
      EQUIVALENCE (INFORM(5),DATE(1)),(INFORM(9),TIME(1))
      COMMON /FILES/ICD,IPT,IPI,IP2,ISCH,IHEAD,IDCT,IQUEST,ISTUD,ISCORE,
     CITEXT,ISYS,IP3,IP4,IP5
      DATA LIC/20*'----'/
      PAGE =1
      LNPI=15
C
C  READ IN THE NUMBER OF REMEDIAL SESSIONS AND THE MAXIMUM NUMBER
C    OF STUDENTS ALLOWED IN A SESSION.
C
      READ(ICD,1)NGMI,MSTUD
    1 FORMAT(2I3)
C
C  READ IN CLASS LOCATION AND PROFESSOR CARDS.
C
      DO 3 I=1,10
      DO 3 J=1,20
      LOC(I,J)=LIC(J)
    3 CONTINUE
      IN=0
    9 READ(ICD,2,END=6) ITEMP
    2 FORMAT(20A4)
      IN=IN+1
      DO 4 J=1,20
    4 LOC(IN,J)=ITEMP(J)
      GO TO 9
    6 IF(IN.LT.NGMI) WRITE(IPT,7)
    7 FORMAT(1H1,'****** ERROR - NUMBER OF GMI SESSIONS EXCEEDS THE NUM
     1BER OF LOCATION DESCRIPTOR CARDS...')
C
C  SORT TC'S FROM MOST TO LEAST MISSED.
C
      DO 5 I=1,200
      IVEX(I,1)=I
    5 IVEX(I,2)=ITCS(I)
      DO 10 I=1,199
      J1=I+1
      DO 10 J=J1,200
      IF (IVEX(I,2).GE.IVEX(J,2)) GO TO 10
      DO 8 K=1,2
      ICNT(K)=IVEX(I,K)
      IVEX(I,K)=IVEX(J,K)
    8 IVEX(J,K)=ICNT(K)
   10 CONTINUE
C
C  CHECK THAT NO. OF T.C.'S NOT LESS THAN NO. OF SESSIONS
C
      IF(IDATA(1).LT.NGMI) NGMI=IDATA(1)
C
C  DETERMINE HOW MANY TC'S ARE TO BE ASSIGNED TO EACH GMI SESSION.
C
      I=IDATA(1)/NGMI
      I1=I*NGMI
      IREM=IDATA(1)-I1
C
C  SET UP NUMBER OF TC'S IN EACH SESSION
```

```
      AIMS III SOURCE STATEMENT LISTING

C
      DO 20 J=1,NGMI
      K=NGMI-J+1
      IF (IREM) 11,11,12
   12 ICNT(K)=I+1
      IREM=IREM-1
      GO TO 20
   11 ICNT(K)=I
   20 CONTINUE
C
C   DETERMINE WHICH TO'S ARE TO BE ASSIGNED TO EACH SESSION.
C
      IREM=IDATA(1)-II
      IK=0
      ICRT=1
   30 DO 25 IP=1,NGMI
      IK=IK+1
      IF(IK.GT.IDATA(1)) GO TO 39
      ICLASS(IP,ICRT)=IK
   25 CONTINUE
      ICRT=ICRT+1
      IF(ICRT.LE.I) GO TO 30
      IM=(NGMI-IREM)+1
      DO 61 IP=IM,NGMI
      IK=IK+1
      IF(IK.GT.IDATA(1)) GO TO 39
      ICLASS(IP,ICRT)=IK
   61 CONTINUE
   39 CONTINUE
C
C   PRINT OUT CLASS ASSIGMENT OF TO'S
C
C
      DO 50 I=1,NGMI
      CALL HEADPG(IRPT,PAGE)
      WRITE(IPT,31) INLES,I,(LOC(I,J),J=1,20)
   31 FORMAT(1H0,//43X,'REMEDIAL SESSION - LESSON PLAN - LESSON ',I3//21
     CX,'SESSION ',I3,5X,'PROF. AND/OR LOCATION  ',1X,20A4//32X,'TERMINA
     CL OBJECTIVE',11X,'CUTOFF LEVEL',8X,'NO. OF STUDENTS'/32X,18('-'),1
     C1X,12('-'),8X,15('-')/)
      K1=ICNT(I)
      DO 40 J=1,K1
      WRITE(IPT,45)IVEX(ICLASS(I,J),1),LEVELS(IVEX(ICLASS(I,J),1)),IVEX(
     CICLASS(I,J),2)
   45 FORMAT(1H0,34X,I4;25X,I4,18X,I5)
   40 CONTINUE
   50 CONTINUE
      CALL CONKER(INLES,IDATA,ITOS,LEVELS,ICLASS,IVEX,NGMI,NSTUD,ICNT,LO
     CC)
      RETURN
      END
      SUBROUTINE CONKER (INLES,IDATA,ITOS,LEVELS,ICLASS,IVEX,NGMI,NSTUD,
     CICNT,LOC)
C
C   THIS ROUTINE WILLS ASSIGN STUDENTS TO THE PROPER REMEDIAL SESSION
C
      DIMENSION ICNT(10),IVEX(200,2),LOC(10,20),ICLASS(10,20),IDATA(5)
      DIMENSION ITOS(200),LEVELS(200),NCLAS(10,100),IREJCT(90)
      INTEGER*2 INFORM(12),DATE(4),TIME(4),RECD(65),STDATA(201),ICTO(11)
      AIMS III SOURCE STATEMENT LISTING

      EQUIVALENCE (INFORM(5),DATE(1)),(INFORM(8),TIME(1))
      COMMON/FILES/ICD,IP1,IPT,IP2,ISCH,IHEAD,IDCT,IGRPST,ISTUD,ISCORE,I
     CTEXT,ISYS,IP3,IP4,IP5
C
C  CREATE RANDOM SEED AND NUMBER TO START SEARCH OF STUDENT FILE
C  IN ORDER TO ASSURE FAIRNESS IN CHOICE OF SECTION
C
C
      IRPT=13
      IPAGE=1
      CALL INFO(INFORM)
      ISEED=DATE(2)+DATE(3)+TIME(4)
      ISEED=IABS(ISEED)
      ISEED=((ISEED/8)*2)+1
      CALL RANDU(ISEED,IY,YFL)
C
C  CALCULATE STARTING STUDENT NUMBER
C
      INUM=YFL*IDATA(2)
      INAM=INUM-1
C
C  ZERO OUT THE CLASS COUNTERS AND STUDENT NUMBERS IN CLASS
C
      DO 5 I=1,NGMI
      DO 4 J=1,NSTUD
    4 NCLAS(I,J)=0
    5 ICTO(I)=0
      N=NGMI+1
      ICTO(N)=0
C
C  READ IN STUDENTS FROM FILE 5 WITH THEIR TO'S
C
      IRT=IDATA(2)-INUM+1
      K1=IDATA(2)
      DO 100 IRCD=1,K1
      IF (IRCD.LE.IRT) GO TO 10
      IV=IV+1
      IND=IV
      GO TO 11
   10 IV=0
      IND=INUM+(IRCD-1)
   11 JSC=(4*IND)-3
      READ(ISCH'JSC)(STDATA(J),J=1,65)
      JSC=JSC+1
      READ(ISCH'JSC)(STDATA(J),J=66,130)
      JSC=JSC+1
      READ(ISCH'JSC)(STDATA(J),J=131,195)
      JSC=JSC+1
      READ(ISCH'JSC)(STDATA(J),J=196,201)
      IF(STDATA(201).EQ.0)GO TO 100
C
C  DETERMINE WHICH SECTION THE STUDENT BELONGS IN BY FINDING TO GROUP
C  HE HAS MISSED MOST ON
C
      ISAVE=0
      ISUM=0
      DO 50 I=1,NGMI
      J1=ICNT(I)
      ISUM=0
      AIMS III SOURCE STATEMENT LISTING

      DO 40 J=1,I1
      K=IVEX(ICLASS(I,J),1)
      ISUM=ISUM+STDATA(K)
   40 CONTINUE
      IF(ISUM.LT.ISUM) GO TO 30
      ISAM=ISUM
      ISAV1=ISAVE
      ISUM=ISUM
      ISAVE=I
      GO TO 50
   30 IF(ISUM.LT.ISAM) GO TO 50
      ISAM=ISUM
      ISAV1=I
   50 CONTINUE
C
C   CHECK THAT CLASS IS NOT FULL
C
      IF(ICTG(ISAVE).GE.NSTUD)GO TO 60
      ICTG(ISAVE)=ICTG(ISAVE)+1
      NCLAS(ISAVE,(ICTG(ISAVE)))=IND
      GO TO 100
C
C   IF FULL TRY SECOND CHOICE
C
   60 IF(ICTG(ISAV1).GE.NSTUD)GO TO 70
      ICTG(ISAV1)=ICTG(ISAV1)+1
      NCLAS(ISAV1,(ICTG(ISAV1)))=IND
      GO TO 100
C
C   SET UP FILE OF DOUBLE REJECTIONS
C
   70 N=NGM1+1
      ICTG(N)=ICTG(N)+1
      IREJCT(ICTG(N))=IND
  100 CONTINUE
C
C  LIST OUT CLASS ROSTERS
C
      DO 200 I=1,NGM1
      CALL HEADPG(IRPT,IPAGE)
      I1=ICNT(I)
      WRITE(IPT,110)INLES,I,(LOC(I,J),J=1,20),(IVEX(ICLASS(I,J),1),J=1,I
     C1)
  110 FORMAT(1H0,43X,'REMEDIAL SESSION ROSTER'/44X,23('-')//47X,'LESSON
     C',I3/47X,10('-')//47X,'SESSION ',I3/47X,11('-')//20X,'PROF AND/OR
     CLOCATION- ',20A4///65X,'TERMINAL OBJECTIVES'/40X,90('-')//10X,'TER
     CMINAL OBJECTIVE NUMBERS-    ',20I4)
      WRITE(IPT,109) (LEVELS(IVEX(ICLASS(I,J),1)),J=1,I1)
  109 FORMAT(1H0,9X,'THRESHOLD LEVELS SET AT    -    ',20I4)
      WRITE(IPT,111)
  111 FORMAT(1H0,'THE FOLLOWING STUDENTS HAVE BEEN ASSIGNED TO THIS REME
     CDIAL SESSION'//7X,'STUDENT NAME',10X,'NUMBER',15X,'PERCENTAGE ERRO
     CR ON THE ABOVE TERMINAL OBJECTIVES'/1X,26('-'),2X,6('-'),5X,80('-'
     C))
      K=ICTG(I)
      IF(K.GT.0) GO TO 312
      WRITE(IPT,220)
  220 FORMAT(1H0,10X,'****     NO STUDENTS ASSIGNED TO THIS SESSION     **
     1**')
```

```
      GO TO 200
  312 DO 150 IV=1,K
  313 CONTINUE
      READ(ISTUD'NCLAS(I,IV))RECD
  308 CONTINUE
      INC=NCLAS(I,IV)
      IBT=(4*INC)-3
      READ(ISCH'IBT)(STDATA(J),J=1,65)
  305 CONTINUE
      IBT=IBT+1
      READ(ISCH'IBT)(STDATA(J),J=66,130)
  307 CONTINUE
      IBT=IBT+1
      READ(ISCH'IBT)(STDATA(J),J=131,195)
      CONTINUE
      IBT=IBT+1
      READ(ISCH'IBT)(STDATA(J),J=196,261)
      CONTINUE
      KT=ICNT(I)
  311 CONTINUE
      WRITE(IPT,215)(RECD(J),J=6,18),NCLAS (I,IV),(STDATA(IVEX(ICLASS(I,
     CJ),I)),J=1,KT)
  215 FORMAT(1H ,13A2,3X,I5,5X,20I4)
  150 CONTINUE
  200 CONTINUE
      N=NGMI+1
      K=ICTC(N)
      IF(K.LT.1) GO TO 260
      CALL HEADPG(IRPT,IPAGE)
      WRITE(IPT,210)INLES
  210 FORMAT(1H0,43X,'REMEDIAL SESSION ROSTER'/44X,23('-')//47X,'LESSON-
     C ',I3/47X,11('-')//20X,      'THE FOLLOWING STUDENTS HAVE NOT BEEN
     CPITTED INTO THEIR FIRST OR SECOND CHOICE'///10X,'ASSIGNMENT TO A
     CSESSION WILL HAVE TO BE DONE BY THE PROFESSOR.'///)
      WRITE(IPT,205)
  205 FORMAT(1H ,15X,'STUDENT NAME',12X,'NUMBER',5X.'SESSION ASSIGNMENT'
     C/8X,26('-'),5X,5('-'),5X,20('-')//)
      DO 250 I=1,K
      K1-IREJCT(I)
      READ(ISTUD'K1) RECD
      WRITE(IPT,211)(RECD(J),J=6,18),IREJCT(I)
  211 FORMAT(1H ,7X,13A2,5X,I5)
  250 CONTINUE
  260 RETURN
      END
/*
   INCLUDE RANDU
   PHASE REPT0012,REPT0C01
// EXEC FFORTRAN
      SUBROUTINE REP11
      IMPLICIT INTEGER*2(K)
      INTEGER*2 CRNM(6),TP(5),RP,LN,ST,SC,NINFIL,NINCRS
      INTEGER*2 RESREC(65),RVOL,RCSN,RSG,RTP
      INTEGER*2 STDBCK(65),OUTCHK(10),STDNME(13)
      INTEGER IOTPT/3/,INPTC/1/,IRESPT/4/,IAIMSY/12/,ISTDBK/9/
      INTEGER OUTPUT(2)/'SUB.','NO/S'/
      INTEGER*2 CSN
      EQUIVALENCE (RVOL, RESREC(1)) , (RCSN,RESREC(2)),(RSG,RESREC(3))
      EQUIVALENCE (RTP,RESREC(4)),(STDNME(1),STDBCK(6)),(CSN,STDBCK(2))
  AIMS III SOURCE STATEMENT LISTING

      EQUIVALENCE (KDROP, STDBCK(24))
      IPGE=1
      INPT=11
      READ(IAIMSY'1) NINFIL,NINCRS
      READ(INPTC,10) RP,LN,ST,SO,(TP(K),K=1,5),(CRNM(K),K=1,6)
   10 FORMAT(I2,7X,3(2,5I1,9X,6A2)
 2070 CONTINUE
      CALL SUBMIT(TP,ST)
      ERRRD=0
      KSTD=1
 2010 READ(IRESPT,END=2000) RESREC
      IF(LN-RVOL) 2030,2020,2010
 2020 CONTINUE
      CALL CHECK(RESN,RTP,RSO)
      GO TO 2010
 2030 ERRRD=-1
      BACKSPACE IRESPT
 2000 CONTINUE
      KPGE= (NINFIL/45)+1
      DO 3000 KD2=1,KPGE
      CALL HEADPG(INPT,IPGE)
      WRITE(IOTPT,510)
  510 FORMAT(  45X,'*** VOLUME SUBMITTAL REVIEW ***')
      WRITE(IOTPT,515) (CRNM(K),K=1,6),LN
  515 FORMAT( /,40X,'COURSE, ',6A2,5X,'VOLUME NO.  ',I3)
      WRITE(IOTPT,550)
  550 FORMAT(/,24X'NOTE,  NO/S - SIGNIFIES THAT STUDENT DID NOT SUBMIT M
     1ATERIALS FOR PROCESSING',
     2/,31X'SUB. - SIGNIFIES THAT STUDENT DID SUBMIT MATERIALS FOR PROCE
     3SSING',/)
      WRITE(IOTPT,530)
  5300FORMAT  (10X,'STUDENT',15X,'CSN  PRE.',5(3X,'STUDY'),
     1'  ASSIGN    HOME    POST',/,
     237X,'TEST   GDE(1)  GDE(2)  GDE(3)  GDE(4)  GDE(5)',
     311X,'WORK    TEST')
      DO 3010 KD3=1,45
      READ(ISTDBK'KSTD)STDBCK
      IF(KDROP.EQ.1)GO TO 1030
      CALL GETCHK(CSN,OUTCHK)
     OWRITE(IOTPT,520) (STDNME(K1),K1=1,13),CSN,(OUTPUT(OUTCHK(K2)),
     1K2=1,9)
      GO TO 1070
 1030 WRITE(IOTPT,540)(STDNME(K1),K1=1,13),CSN
 1070 IF(KSTD.GE.NINFIL) GO TO 3000
      KSTD=KSTD+1
 3010 CONTTNUE
 3000 CONTINUE
      IF(ERRRD.EQ.0) GO TO 9999
      LN=RVOL
      GO TO 2070
  520 FORMAT(1X,12A2,A1,6X    ,I3,2X,9(A4,4X))
  540 FORMAT(1X,12A2,A1,6X,I3,'   STUDENT DROPPED')
 9999 RETURN
      END
      SUBROUTINE REP12
      INTEGER*2 STDBCK(65),STDNME(13),CRNM(6),NINFIL,NINCRS
      EQUIVALENCE (STDNME(1),STDBCK(6))
      DATA (NPTC,IOTPT,ISTDB,IAIMSY,IPGE/1,3,9,12,1/
      JSTD=1
```

```
        PIAS III SOURCE STATEMENT LISTING

        READ(INPTC,10)RP,(CRNM(K),K=1,6)
        READ(IAIMSY'1)NINFIL,NINCRS
        KDUM2=(NINFIL/45)+1
        DO 1000 KDUM1=1,KDUM2
        CALL HEADPG(RP,IPGE)
        WRITE(IOTPT,520)(CRNM(K),K=1,6)
        WRITE(IOTPT,530)
        DO 1100 KDUM3=1,45
        READ(ISTDU'JSTD)STDDCK
       WRITE(IOTPT,510)(STDNME(K2),K2=1,13),(STDDCY(K3),K3=14,23),STDDCK(
       12)
        IF(JSTD.GT.NINFIL) GO TO 1000
        JSTD=JSTD+1
  1100 CONTINUE
  1000 CONTINUE
    10 FORMAT(I2,27X,6A2)
   510 FORMAT(1X,12A2,A1,5X,5A2,5X,I4)
   520 FORMAT(55X,'*** CLASS ROSTER ***',/,55X,'COURSE, ',6A2,//)
   530 FORMAT(10X,'STUDENT',14X,'ACADEMY'8X,'COURSE',/,33X,'NO.',10X,
       1'STD.NO.')
        RETURN
        END
 /*
    PHASE REPT0013,REPT0001
 // EXEC FFORTRAN
        SUBROUTINE REP14
        IMPLICIT INTEGER*2 (K)
        INTEGER*2 RLN,ERRG
        INTEGER*2 NINFIL,NINCRS
        INTEGER*2 CRNM(6:,TP(5),IL
        INTEGER*2 RP,LN,ST,SC,CTQT,EH,JSTD,KDROP,TEST T,STDNO
        INTEGER*2 STDNME(13),OUTPUT(15)
        EQUIVALENCE (JSTD,STDNO)
        INTEGER INPTC/1/,IOTPT/3/,IRESPT/4/,IAIMSY/12/
        OUTC=0
        JSTD=1
        IRPT=14
        IPGE=1
        KIL=0
 C                                                    SRCNIN,LIFNIN )1'YS.IAI(D
        NINFIL=10
        NINCRS=10
        READ(INPTC,10) RP,LN,ST,SC,(TP(J),J=1,5),(CRNM(J1),J1=1,6)
  2020 CONTINUE
        CALL REDREC(RP,LN,ST,'',CTQT,EH)
        IF(EH.NE.0) GO TO 999
        CALL SUBMIT(TP,ST)
        CALL CUMAVE
        CALL GETREC(RP,LN,RLN,NINFIL,NINCRS,ERRG)
        CALL GRPST
        CALL REP13(CRNM)
        JDUM2=(NINFIL/45)+1
        DO 1100 JDUM1=1,JDUM2
        CALL HEADPG(IRPT,IPGE)
        WRITE(IOTPT,510)
        WRITE(IOTPT,515)(CRNM(K),K=1,6),LN
        WRITE(IOTPT,501)
        WRITE(3,530)
        DO 1000 JDUM3=1,45
```

FILMED FROM BEST AVAILABLE COPY

AIMS III SOURCE STATEMENT LISTING

```
      CALL GETRES(JSTD,STDNAM,OUTPUT,KDROP,ILSTCT)
      IF(KDROP.EQ.1) GO TO 1050
      CALL GETCUM(OUTPUT)
      WRITE(IOTPT,520)(STDNAM(K4),K4=1,13),STDNO,(OUTPUT(K4),K4=1,13)
      GO TO 1070
 1050 WRITE(IOTPT,540)(STDNAME(K5),K5=1,13),STDNO
      GO TO 1070
 1040 WRITE(IOTPT,550)(STDNAME(K5),K5=1,13),STDNO
 1070 IF(JSTD.GE.NTAPE(L)) GO TO 1100
      JSTD=JSTD+1
 1000 CONTINUE
 1100 CONTINUE
      IF(ERRG) 2000,2000,2010
 2000 LN=FLN
      GO TO 2020
 2010 CONTINUE
   10 FORMAT(I2,7X,3I2,5I1,9X,6A2)
   13 FORMAT(62X,I2)
   14 FORMAT(10(I5,I3))
  501 FORMAT(1H0)
  510 FORMAT( 39X,'*** INDIVIDUAL VOLUME STATISTICS ***')
  515 FORMAT( 35X,'COURSE, ',6A2,10X,'VOLUME NO. ',I3)
  520 FORMAT(1X,12A2,A1,8X      ,I3,13(2X,I4,'.'))
  5300FORMAT( 53X,'ABS.   REL.',10X,'POST',10X,'ABS.   ALL.   CM.AV. ',
     1'CM.AV. CM.AV. CM.AV.',/,
     210X'STUDENT',17X,'CSN  CAPB.  PERF.  PERF.  PERF.  PROB.  TEST'
     3,'   NET    ACH.   ACH.   PROB.  PT.TS. NET    REL.',/,
     439X,'INDEX  INDEX  DEV.   DEV.   ACH.   ACH.   ACH.   DEV.    ',
     5'DEV.   ACH.   ACH.   ACH.   ACH.DEV.')
  540 FORMAT(1X,12A2,A1,I3,' STUDENT DROPPED')
  5500FORMAT(1X,12A2,A1,I3,6X,'THIS STUDENT DOES NOT HAVE ANY TESTS OR M
     1ATERIAL TO BE GRADED')
  999 RETURN
      END
      SUBROUTINE CUMAVE(NF)
      INTEGER*2 OUTPUT(13),CUMSL(4),CUMNL(4),NF,CSN
      INTEGER*2 LN,CNUM(6)
      INTEGER INPTC/1/,ISCRAT/5/,IPUNH/2/
      READ(INPTC,15) LN
   15 FORMAT(70X,I5)
      DO 1000 KCSN=1,NF
      READ(INPTC,10) CSN,(CUMSL(K),CUMNL(K),K=1,4)
   10 FORMAT(I3,4(I5,I3))
      JSN=CSN/1
      WRITE(ISCRAT'JSN) CSN,(CUMSL(K),CUMNL(K),K=1,4)
 1000 CONTINUE
      RETURN
      ENTRY GETCUM(CSN,OUTPUT)
      JSN=CSN/1
      READ(ISCRAT'JSN) CSN,(CUMSL(K),CUMNL(K),K=1,4)
      DO 2000 K=1,3
      CUMSL(K)= CUMSL(K) + OUTPUT(K+4)
      CUMNL(K)= CUMNL(K) + 1
 2000 CONTINUE
      CUMSL(4)= CUMSL(4) + OUTPUT(9)
      CUMNL(4)= CUMNL(4) + 1
      DO 2010 K=10,13
      OUTPUT(K)=CUMSL(K-9)/CUMNL(K-9)
      WRITE(ISCRAT'JSN) CSN,(CUMSL(K),CUMNL(K),K=1,4)
      RETURN
      ENTRY OUTCUM(LN,CPNM)
      WRITE(IPUNH,20)(CPNM(K1),K1=1,6),LN
  20 FORMAT(' HEADER RECORD FOR CUM. AVE. REPORT 14 COURSE ',6A2,
     1' VOLUME NO. ',I3)
      DO 3000 JNF=1,NF
      READ(ISCRAT'JNF) CSN,(CUMSL(K),CUMNL(K),K=1,4)
      WRITE(IPUNH,15) CSN,(CUMSL(K),CUMNL(K),K=1,4)
 3000 CONTINUE
      RETURN
      END
      SUBROUTINE GETREC(RP,LN,RLN,NF,NC,CTOT,EG)
      INTEGER*2 RP,NF,NC,CTOT,LN
      INTEGER*2 GS(185,11),GC(185)
      INTEGER*2 GRAD,RLN,CSN,RSG,RTP,EG
      INTEGER*2 OUTCHK(10),KST
      INTEGER*2 CPNM(6)
      INTEGER*2 GID(5)/1,3,5,10,11/
      DO 1000 KD1=1,NF
      GC(KD1)=0
      DO 1000 KD2=1,11
      GS(KD1,KD2)=0
 1000 CONTINUE
 2000 CONTINUE
      CALL GRADE(GRAD,RLN,CSN,RSG,RTP,EG)
      IF(EG.NE.0) GO TO 2010
      CALL CHECK(CSN,RTP,RSG)
      IND=GID(RTP+2*(RSG-1))
      GS(CSN,IND)=GRAD
      GO TO 2000
 2010 CONTINUE
      DO 2200 KST=1,NF
      CALL GETCHK(KST,OUTCHK)
      DO 2300 KD1=1,9
      IF(OUTCHK(KD1).EQ.1) GC(KST)=GC(KST)+1
 2300 CONTINUE
      IF(GC(KST).NE.CTOT.AND.GC(KST).NE.0) GC(KST)=-2
 2200 CONTINUE
   10 FORMAT(1X,20I6)
  999 RETURN
      ENTRY GRPST
C VARIABLE ALLOCATION FOR GROUP STATISTICS
      INTEGER   GRPSTT(7,4),JD(7)
      INTEGER*2 STONME(165,13),STDBCK(65),SN(13),CAPIN,STDNO
      EQUIVALENCE (STDNO,STDBCK(2)),(CAPIN,STDBCK(25)),(SN,STDBCK(6))
      WT=3
      NGRPC=0
      DO 3000 NGT=1,7
      GRPSTT(NGT,1)= 1000
      GRPSTT(NGT,2)= 0
      GRPSTT(NGT,3)=-1000
      GRPSTT(NGT,4)= 0
 3000 CONTINUE
      IBCK=9
        DO 2110 K9=1,NF
        READ(IBCK'K9)STDBCK
        IF(STDBCK(24).EQ.1) GC(STDNO)=-1
```

AIMS III SOURCE STATEMENT LISTING

```
      GS(JSTNO,1)=CAPIN
      DO 2120 K6=1,13
      STORE(JSTNO,K6)= SM(K6)
 2120 CONTINUE
 2110 CONTINUE
      DO 2100 KSTD=1,NF
      IF(OC(KSTD).EQ.-2.OR.OC(KSTD).EQ.0) GO TO 2100
      GS(KSTD,2) = (WT*GS(KSTD,10)+GS(KSTD,11))/(WT+1)
      GS(KSTD,7)= (GS(KSTD,2)+GS(KSTD,5))/2
      GS(KSTD,3)=GS(KSTD,2)-GS(KSTD,1)
      GS(KSTD;8)=GS(KSTD,7)-GS(KSTD,1)
      DO 3010 NGT=1,7
      JD(NGT)=GS(KSTD,NGT)/1
 3010 CONTINUE
      DO 3020 NGT=1,7
      GRPSTT(NGT,1)=MIN0(GRPSTT(NGT,1),JD(NGT))
      GRPSTT(NGT,3)=MAX0(GRPSTT(NGT,3),JD(NGT))
      GRPSTT(NGT,4)= GRPSTT(NGT,4)+JD(NGT)
 3020 CONTINUE
      NGRPC=NGRPC+1
 2100 CONTINUE
      IF(NGRPC.EQ.0) GO TO 3050
      DO 3030 NGT=1,7
      GRPSTT(NGT,2) = GRPSTT(NGT,4)/NGRPC
 3030 CONTINUE
      GRPSTT(3,2)= GRPSTT(2,2)-GRPSTT(1,2)
      GRPSTT(4,2)= GRPSTT(7,2)-GRPSTT(1,2)
      DO 3040 KSTD=1,NF
      GS(KSTD,4)=GS(KSTD,3)-GRPSTT(3,2)
      GS(KSTD,9)=GS(KSTD,8)-GRPSTT(4,2)
 3040 CONTINUE
 3050 RETURN
      ENTRY REP13(CRNM)
      IRPT=15
      IPAG=1
      IOTPT=3
      CALL HEADPG(IRPT,IPAG)
      WRITE(IOTPT,510)(CRNM(K),K=1,6),LN
  510CFORMAT(20X,'*** VOLUME STATISTICS ***',/,10X,'CCURSE, ',6A2,
     110X,'VOLUME NO.  ',I3,//,23X,'MINIMUM',07X,'MEAN',08X,'MAXIMUM')
     CWRITE(IOTPT,520) (GRPSTT(1,KD),KD=1,3),
     1                  (GRPSTT(2,KD),KD=1,3),
     1                   GRPSTT(3,2),
     1                  (GRPSTT(5,KD),KD=1,3),
     1                  (GRPSTT(6,KD),KD=1,3),
     1                   GRPSTT(7,2),
     1                   GRPSTT(4,2)
  520CFORMAT(1X,'CAPABILITY INDEX',            T26,3(I3,10X),/,
     1         1X,'PERFORMANCE INDEX',          T26,3(I3,10X),/,
     2         1X,'PERFORMANCE DEVIATION',      T39,I3,/;
     3          1X'PROBLEM ACHIEVEMENT'         T26,3(I3,10X),/,
     4          1X'POST TEST ACHIEVEMENT',      T26,3(I3,10X),/,
     5         1X,'NET ACHIEVEMENT INDEX',      T39,I3,/,
     6         1X,'ACHIEVEMENT DEVIATION',      T39,I3)
      RETURN
      ENTRY GETRES(JSTD,SM,OUTPUT,KDRCP, TESTCT)
      INTEGER*2 OUTPUT(13),JSTD,KDRCP,TESTCT,SM(13)
      DO 4000 K1=1,9
```

AIMS III SOURCE STATEMENT LISTING

```
 4000 CONTINUE
      DO 4010 KI=1,15
      SM(KI)=STDNMF(JSTD,KI)
 4010 CONTINUE
      KDRGP=0
      IF(CC(JSTD).EQ.-1) KDRGP=1
      TESTCT=CC(JSTD)
      RETURN
      END
      SUBROUTINE GRADE(GRAD,RLN,CSN,RSG,RTP,EG)
C THIS SECTION OF SUBR GRADE READS A RESP. RECORD SEQUENCIALY
C AND GRADES THE TEST
      IMPLICIT INTEGER*2 (K)
      INTEGER FILES(5)/12,6,3,4,0/,IRESPT/4/,IQTPT/3/
      INTEGER*2 HDRC(12,48),HR(65),RESPR(65),ANS(48),NUMQT(12),IP(5)
      INTEGER*2 GRAD,CSN,RSG,RTP,EG,LN,ST,CTGT,EH,RLN
      INTEGER*2 COUNT,TPE,QNO,RIGHT,ITEST,SG,SM
      INTEGER*2 SUBGRD,SUMGRD,NUMRSP(12),HDRCD(48)
      EQUIVALENCE (RESPR(18),ANS(1)),(QNO,HR(13))
      EG=0
      GRAD=0
      COUNT=0
C READ RESPONSE TAPE
 3000 READ(IRESPT,END=9999)RESPR
      RLN=RESPR(1)
C TEST FOR PROPER VOL. NO. (LN)
      IF(LN-RLN) 3020,3010,3000
 3010 CONTINUE
      CSN=RESPR(2)
      RSG=RESPR(3)
      RTP=RESPR(4)
C THIS EQUATION MAPS A 2-D FIELD INTO A 1-D FIELD
      TPE= RTP+2*(RSG-1)
      QNO=NUMQT(TPE)
C CALCULATE A NUMERIC GRADE FOR TEST
      DO 1000 K=1,QNO
C FUNCTION SUBGRD GRADES RESP. USING APPROP.  ALGO.
      ITEST= SUBGRD(RTP,HDRC(TPE,K),ANS(K))
      COUNT=COUNT+ITEST
 1000 CONTINUE
      GRAD = COUNT/QNO
      IF(KP.EQ.15.OR.RTP.NE.4) GO TO 999
      DO 3030 K=1,QNO
      HDRCD(K)=HDRC(TPE,K)
 3030 CONTINUE
      GRAD = SUMGRD(QNO,NUMRSP(TPE),HDRCD,ANS)
  999 RETURN
C EG=1 IS RETURN CODE FOR END OF FILE
 9999 EG=1
      RETURN
 3020 CONTINUE
C EG=-1 RETURN CODE DIFFERENT VOL. NO.
      EG=-1
      LN=RLN
      BACKSPACE  IRESPT
      RETURN
C
C
```

AIMS III SOURCE STATEMENT LISTING

```
      WRITE(IOTPT,515)
C
C THIS SECTION PICKS OUT THE HEADERS REQUIRED TO GRADE THE TEST
C AND STORES THEM IN THE ARRAY CALLED  HDRC
C
      DO 2010 K2=1,12
      NUMQT(K2)=1
      DO 2010 K3=1,48
      HDRC(K2,K3)=0
 2010 CONTINUE
      CTQT=0
C THIS LOOP FETCHS HEADER RECORD AND STORES IT
C     PLUS COUNTS TOTAL EXAMS TO BE GRADED (CTQT)
C     AND NUMBER OF QUESTIONS PER TEST (NUMQT)
      DO 1500 KT=1,5
      SG=1
      IF(TP(KT).LE.0.OR.TP(KT).GT.5) GO TO 1500
      IF(TP(KT).NE.4) GO TO 1530
 1520 DO 1540 SR=1,ST
      SG=SR
 1530 CALL GETIT(HR,LN,SG,TP(KT),FILES,EH)
      IF(EH.NE.0) WRITE(IOTPT,510) EH,LN,TP(KT),SG
      TPE=TP(KT)+2*(SG-1)
      DO 1550 K1=1,48
      HDRC(TPE,K1)= HR(17+K1)
 1550 CONTINUE
      CTQT=CTQT+1
      NUMQT(TPE)=QNO
      NUMRSP(TPE)=HR(11)
      IF(TP(KT).NE.4) GO TO 1500
 1540 CONTINUE
 1500 CONTINUE
      RETURN
  510 FORMAT(' *** WARNING *** ERROR IN OBTAINING HEADER RECORD,  ERROR,
     1',I2,' LESSON.',I2,' TYPE.',I2,' SEGMENT.',I2)
  515 FORMAT(1H1)
      END
      FUNCTION SUMGRD(Q,N,H,A)
      INTEGER*2 Q,H(48),A(48),N,K
      INTEGER*2 IPASS,SUMGRD,ICORR,NP,D(11),SCORR,SNP,INCORR
   10 FORMAT(40I3)
      SCORR=0
      SNP=0
      DO 1000 K=1,Q
      ICORR=IPASS(H(K),A(K),NP,D)
      SCORR=SCORR+ICORR
      SNP=SNP+NP
 1000 CONTINUE
      INCORR=SNP-SCORR
      SUMGRD= (SCORR*100-(Q*INCORR*100 / (N-Q)))/Q
      RETURN
      END
/*
   PHASE REPT0015,REPT0001
// EXEC FFORTRAN
      SUBROUTINE REP15
      IMPLICIT INTEGER*2 (K)
      INTEGER*2 NINFIL,NINCRS
      [illegible]
```

FILMED FROM BEST AVAILABLE COPY

```
        AIMS III SOURCE STATEMENT LISTING

        INTEGER*2 STUBCK(65),STUNME(13),CRNM(6),TP(5)
        INTEGER*2 OUTCTN(10),CTPT,NHW,NPT,CAPIN
        INTEGER*2 RP,LN,ST,SC,GRAD,RLN,CSN,RSG,RTP,EG,EH,IND,CTQT,LLN
        INTEGER*2 SECT(2),GROP(2),STUNO,COMAV
        INTEGER*2 SUBOUT(2)/'  ','***'/
        INTEGER*2 KSTD,PTDUM(2),HWDUM(2),ERRCOM
        INTEGER*2 UID(12)/1,9,8,2,7,3,10,4,10,5,10,6/
        INTEGER INPTC/1/,ICTPT/3/,IRESPT/4/,IAIMSY/12/
       EQUIVALENCE (STUNME(1),STUBCK(6)),(STUNO,STUBCK(2))
        EQUIVALENCE (KENCP,STUBCK(24)),(CAPIN,STUBCK(25))
        EQUIVALENCE (SECT(1),STUBCK(42)),(GROP(1),STUBCK(44))
        IRPT=15
C READ IN THE NO. OF STUDENTS ENROLLED IN THE COURSE.
        READ(IAIMSY'1) NINFIL,NINCRS
C READ IN THE REPORT REQUEST CARD
        READ(INPTC,10) RP,LN,ST,SC,(TP(J),J=1,5),(CRNM(J1),J1=1,6),IGRPT
        IF(LN.EQ.1) CALL COMOIS(NINFIL)
C THIS LOOP RETRIEVES COMM. AVE. DATA FROM SCORE FILE
        DO 3010 KSTD = 1,NINFIL
        CALL GETCMM(KSTD,LN,PTDUM,HWDUM,ERRCOM)
        IF (ERRCOM.EQ.-1) GO TO 4000
        DO 3010 L1=1,2
        PTLT(KSTD,L1) = PTDUM(L1)
        HWLT(KSTD,L1) = HWDUM(L1)
  3010 CONTINUE
  2070 CONTINUE
        CALL RSEP(RP)
        JSTD=1
        IPGE=1
C ZERO ARRAYS AS REQUIRED.
        DO 2020 KDUM7=1,NINFIL
        OUTCTN(KDUM7)=0
        DO 2020 KDUM8=1,10
        OUTPUT(KDUM7,KDUM8)=0
  2020 CONTINUE
C SET UP HEADER ARRAY
        CALL HEDREL(LN,ST,TP,CTQT,EH)
C SET UP FOR STUDENT SUBMITTAL CHECK
        CALL SUBMIT(TP,ST)
        KREAD=NINFIL*CTQT
C THIS LOOP READS IN STUDENT RESPONSE TAPE,
C       ALSO GRADES THE TEST AND STORES RESULT (OUTPUT(CSN,IND).
        DO 2000 KDUM4=1,KREAD
        CALL GRAD1(GRAD,RLN,CSN,RSG,RTP,EG)
        IF(EG.NE.0) GO TO 2010
C CHECK ENTRY POINT IN SUBMIT, HERE WE INDICATE THAT STUD, HAS HANDED IN
        CALL CHECK(CSN,RTP,RSG)
C MAP FUNCTION FOR OUTPUT ARRAY
        IND=UID(RTP+2*(RSG-1))
        OUTPUT(CSN,IND)=GRAD
        OUTCTN(CSN)=OUTCTN(CSN)+1
  2000 CONTINUE
  2010 CONTINUE
        JDUM2=(NINFIL/45)+1
C THIS BEGINS THE LOOP FOR THE OUTPUT OF GRADES
        DO 1100 JDUM1=1,JDUM2
        CALL HEADPG(IRPT,IPGE)
        WRITE(ICTPT,510)
```

```
      WRITE(IOTPT,501)
      WRITE(IOTPT,530)
      DO 1000 JOUM3=1,45
C READ STUDENT BAAKGROUND FILE
      READ(9'JSTD)STDBCK
      IF(KDROP.EQ.1) GO TO 1030
      IF(OUTCTN(STDNO).LE.0) GO TO 1040
C FIND OUT WHAT THE STUDENT HAS SUBMITTED FOR PROCESSING
      CALL GETCHK(STDNO,OUTCHK)
      CTPI=0
      SUMAVE=0
C CALCULATE THE SUM OF ALL GRADE
      DO 1060 KAVE=2,8
      IF(OUTCHK(KAVE).EQ.1) CTPI=CTPI+1
C CALCULATE PREF. INDEX (SUMAVE)
      SUMAVE=SUMAVE+OUTPUT(STDNO,KAVE)
 1060 CONTINUE
      IF(CTPI.EQ.0) CTPI=1
      OUTPUT(STDNO,10)=SUMAVE/CTPI
      PTLT(STDNO,1)= PTLT(STDNO,1)+OUTPUT(STDNO,9)
      HWLT(STDNO,1)= HWLT(STDNO,1)+OUTPUT(STDNO,8)
C CALCULATE THE REQUIRED AVERAGES
      CUMAV=0
      NPT=1
      NHW=1
      IF(OUTCHK(8).EQ.2) NHW=0
      IF(OUTCHK(9).EQ.2) NPT=0
      HWLT(STDNO,2)=HWLT(STDNO,2)+NHW
      PTLT(STDNO,2)=PTLT(STDNO,2)+NPT
C CALCULATE CUM.AVE. BELOW
     0IF(HWLT(STDNO,2).EQ.0.AND.PTLT(STDNO,2).GT.0)
     1CUMAV =        PTLT(STDNO,1)/PTLT(STDNO,2)
     0IF(PTLT(STDNO,2).EQ.0.AND.HWLT(STDNO,2).GT.0)
     1CUMAV =        HWLT(STDNO,1)/HWLT(STDNO,2)
      IF(HWLT(STDNO,2).GT.0.AND.PTLT(STDNO,2).GT.0)
     1CUMAV =  3*HWLT(STDNO,1)/(HWLT(STDNO,2)*10)+
     2         7*PTLT(STDNO,1)/(PTLT(STDNO,2)*10)
C OUTPUT NORMAL LINE OF STATISTICS
     0WRITE(IOTPT,520) (STDNME(K4),K4=1,13),SECT(1),SECT(2),GROP(2),
     2STDNO,
     1(OUTPUT(STDNO,K4),SUBOUT(OUTCHK(K4)),K4=1,10),CUMAV,CAPIN
     0IF(IGRPT.NE.0)
     1WRITE(IGRPT,521)LN,(STDNME(K),K=1,13),SECT(1),SECT(2),GROP(2),
     2STDNO,(OUTPUT(STDNO,K4),SUBOUT(OUTCHK(K4)),K4=1,10),CUMAV,
     3SUBOUT(1),CAPIN,SUBOUT(1)
      GO TO 1070
C OUTPUT STUD. DROPPED LINE
 1030 WRITE(IOTPT,540)(STDNME(K5),K5=1,13),STDNO
      GO TO 1070
 1040 WRITE(IOTPT,550)(STDNME(K5),K5=1,13),STDNO
 1070 IF(JSTD.GE.NINFIL) GO TO 1100
      JSTD=JSTD+1
 1000 CONTINUE
 1100 CONTINUE
      IF(EG) 2040,2040,2060
 2040 CONTINUE
      LN=RLN
      GO TO 2070
```

```
        AIFS III SOURCE STATEMENT LISTING

C THIS LOOP LOADS UPDATED CUMM. AVE. DATA BACK ON SCORE FILE
      DO 3020 KSTD=1,NINFIL
      DO 3030 L1=1,2
      PTDUM(L1) = PTLT(KSTD,L1)
      HWDUM(L1) = HWLT(KSTD,L1)
 3030 CONTINUE
      CALL LODCUM(KSTD,LN,PTDUM,HWDUM)
 3020 CONTINUE
      IF(IGRPT.NE.0) END FILE IGRPT
      GO TO 999
 4000 WRITE(IOTPT,5000) LN,LN
 5000CFORMAT(' *** ERROR *** VOLUME NO.,',I3,' HAS BEEN PROCESSED BY REP
     1ORT 15. REPORT REQUEST TERMINATED.',/,20X,'TO GENERATE REPORT 15
     2FOR VOLUME NO.,',I3,' REQUEST A REPORT FOR VOLUME ONE')
   10 FORMAT(I2,7X,3I2,5I1,9X,6A2,8X,I2)
   15 FORMAT(62X,I2)
   14 FORMAT(10(I5,I3))
  501 FORMAT(1H0)
  510 FORMAT(/,50X,'*** VOLUME SUMMARY ***')
  515 FORMAT(//,40X,'COURSE, ',6A2,5X,'VOLUME NO.  ',I3,/,20X,'NOTE,
     1THE ** SIGNIFIES THAT NO MATERIALS HAVE BEEN SUBMITTED FOR PROCESS
     1ING').
  520 FORMAT(1X,12A2,A1,3A2    ,I3,10(2X,I3,'.',A2),2X,I3,'.',4X,I3,'.')
  521 FORMAT(I2,12A2,A1,3A2,I3,12(I3,'.',A2))
  530CFORMAT  (10X,'STUDENT',15X,'CSN  PRE.',5(3X,'STUDY'),
     1'  ASSIGN    HOME     POST    PERF.     CUM.    CAPL.',/,
     237X,'TEST   GDE(1)  GDE(2)  GDE(3)  GDE(4)  GDE(5)',
     311X,'WORK     TEST    INDEX     AVE.    INDEX')
  540 FORMAT(1X,12A2,A1,6X,I3,6X,'STUDENT DROPPED')
  550 FORMAT(1X,12A2,A1,6X,I3,6X,'THIS STUDENT DOES NOT HAVE ANY MATERIA
     1LS TO BE PROCESSED')
  999 RETURN
      END
      SUBROUTINE CUM015(NINFIL)
      INTEGER*2 NINFIL,ZERO/0/,CSN,LN,ERRCUM,VOL
      INTEGER*2 PTLT(2),HWLT(2),DUMMY(65)
      ISCORE=10
C THIS SECTION ZERO'S THE SCORE FILE FOR FRESH START
      DO 10 JSTD=1,NINFIL
      CSN= JSTD/1
      READ(ISCORE'JSTD) DUMMY
      WRITE(ISCORE'JSTD)  CSN,ZERO,ZERO,ZERO,ZERO,ZERO,(DUMMY(I),I=7,65)
   10 CONTINUE
      RETURN
C THIS ENTRY FINDS CUM. AVERAGE DATA
C     ALSO CHECKS FOR OVER LAPPING REPORT REQUEST
      ENTRY GETCMM(CSN,LN,PTLT,HWLT,ERRCUM)
      ISCORE=10
      JCSN=CSN/1
      READ(ISCORE'JCSN)  CSN,VOL,PTLT,HWLT
      ERRCUM=0
      IF(VOL.GE.LN) ERRCUM=-1
      RETURN
C THIS ENTRY LOADS UPDATED CUMM. AVERAGE DATA BACK ON FILE
      ENTRY LODCUM(CSN,LN,PTLT,HWLT)
      JCSN = CSN/1
      READ(ISCORE'JCSN) DUMMY
      WRITE(ISCORE'JCSN) CSN,LN,PTLT,HWLT,(DUMMY(I),I=7,65)
```

```
          AIMS III SOURCE STATEMENT LISTING

       END
       SUBROUTINE GRADE(GRAD,RLN,CSN,RSG,RTP,EG)
C THIS SECTION OF SUBR GRADE READS A RESP. RECORD SEQUENTIALY
C AND GRADES THE TEST
       IMPLICIT INTEGER*2 (K)
       INTEGER FILES(5)/12,6,8,4,0/,IRESPT/4/,IQTPT/5/
       INTEGER*2 HDRC(16,48),HR(65),RESPR(65),ANS(48),NUMQT(16),TP(5)
       INTEGER*2 GRAD,CSN,RSG,RTP,EG,LN,ST,CTQT,EH,RLN
       INTEGER*2 COUNT,TPE,QNO,RIGHT,ITEST,SC,SM
       INTEGER*2 SUBGRD
       EQUIVALENCE (RESPR(18),ANS(1)),(QNO,HR(13))
       EG=0
       GRAD=0
       COUNT=0
C READ RESPONSE TAPE
 3000 READ(IRESPT,END=9999)RESPR
       RLN=RESPR(1)
       IF(LN-RLN) 3020,3010,3000
 3010 CONTINUE
       CSN=RESPR(2)
       RSG=RESPR(3)
       RTP=RESPR(4)
C THIS EQUATION MAPS A 2-D FIELD INTO A 1-D FIELD
       TPE= RTP+2*(RSG-1)
       QNO=NUMQT(TPE)
C CALCULATE A NUMERIC GRADE FOR TEST
       DO 1000 K=1,QNO
C FUNCTION RIGHT CORRECTS IND. QUESTIONS. (1=CORRECT,0=WRONG)
       ITEST= SUBGRD(RTP,HDRC(TPE,K),ANS(K))
       COUNT=COUNT+ITEST
 1000 CONTINUE
       GRAD = COUNT/QNO
  999 RETURN
C EG=1 IS RETURN CODE FOR END OF FILE
 9999 EG=1
       RETURN
 3020 CONTINUE
       EG=-1
       LN=RLN
       BACKSPACE  IRESPT
       RETURN
C
C
       ENTRY HEDREI(LN,ST,TP,CTQT,EH)
       WRITE(IQTPT,515)
C
C THIS SECTION PICKS OUT THE HEADERS REQUIRED TO GRADE THE TEST
C AND STORES THEM IN THE ARRAY CALLED  HDRC
C
       DO 2010 K2=1,16
       NUMQT(K2)=1
       DO 2010 K3=1,48
       HDRC(K2,K3)=0
 2010 CONTINUE
       CTQT=0
C THIS LOOP FETCHS HEADER RECORD AND STORES IT
C      PLUS COUNTS TOTAL EXAMS TO BE GRADED (CTQT)
C      AND NUMBER OF QUESTIONS PER TEST (NUMQT)
```

```
       AIRS III SOURCE STATEMENT LISTING .

      SG=1
      IF(TP(KT).LE.0.OR.TP(KT).GT.5) GO TO 1500
      IF(TP(KT).NE.4) GO TO 1530
 1520 DO 1540 SM=1,ST
      SG=SM
 1530 CALL GETIT(HR,LN,SG,TP(KT),FILES,EH)
      IF(EH.NE.0) WRITE(IOTPT,510) EH,LN,TP(KT),SG
      TPE=TP(KT)+2*(SG-1)
      DO 1550 K1=1,48
      HDRC(TPE,K1)= HR(17+K1)
 1550 CONTINUE
      CTGT=CTGT+1
      NUMGT(TPE)=GNO
      IF(TP(KT).NE.4) GO TO 1500
 1540 CONTINUE
 1500 CONTINUE
      RETURN
  510 FORMAT(' *** WARNING *** ERROR IN OBTAINING HEADER RECORD,  ERROR,
     1',I2,' LESSON.',I2,' TYPE.',I2,' SEGMENT.',I2)
  515 FORMAT(1H1)
      END
      SUBROUTINE REP16
      IMPLICIT INTEGER*2 (K)
      INTEGER*2 SUBOUT(12),STDNME(13),CRNM(6),TP(5)
      INTEGER  OUTPUT(12),CSN
      INTEGER*2 ID(2,2),LID(2,2),LLN
      INTEGER*2 RP,LN,ST,SC
      INTEGER INPTC/1/,IOTPT/3/,IRESPT/4/,IAIRSY/12/,IGRPT
      DATA ID(2,1)/'  '/,LID(2,1)/'  '/
      IRPT=16
C READ IN THE REPORT REQUEST CARD
      READ(INPTC,10) RP,LN,ST,SC,(TP(K),K=1,5),(CRNM(K),K=1,6),IGRPT
C THIS BEGINS THE LOOP FOR THE OUTPUT OF GRADES
     OREAD(IGRPT,521)
     1LN ,(STDNME(K),K=1,13),(ID(1,K),K=1,2),ID(2,2) , CSN,
     2(OUTPUT(K),SUBOUT(K),K=1,12)
 3000 CONTINUE
      LLN=LN
      DO 6000 J=1,2
      DO 6000 K=1,2
      LID(J,K)=ID(J,K)
 6000 CONTINUE
      BACKSPACE IGRPT
      IPGE=1
 5000 CONTINUE
      CALL HEADPG(IRPT,IPGE)
      WRITE(IOTPT,510)
      WRITE(IOTPT,515)(CRNM(K2),K2=1,6),LN
      WRITE(IOTPT,501)
      WRITE(IOTPT,530)
      DO 1000 JDUM3=1,45
C READ STUDENT BACKGROUND FILE
     OREAD(IGRPT,521,END=999)
     1LN ,(STDNME(K),K=1,13),(ID(1,K),K=1,2),ID(2,2) , CSN,
     2(OUTPUT(K),SUBOUT(K),K=1,12)
      IF(LLN.NE.LN) GO TO 3000
      DO 4000 K=1,2
      IF(ID(SC,K).NE.LID(SC,K)) GO TO 3000
 4000 CONTINUE
```

```
         AIMS III SOURCE STATEMENT LISTING

      LLN=LN
      DO 2000 J=1,2
      DO 2000 K=1,2
      LID(J,K)=ID(J,K)
 2000 CONTINUE
C OUTPUT NORMAL LINE OF STATISTICS
      WRITE(IOTPT,520)
     1     (STNAME(K),K=1,13),(ID(1,K),K=1,2),ID(2,2) , CSN,
     2(OUTPUT(K),SCREOT(K),K=1,12)
 1000 CONTINUE
      GO TO 5000
   10 FORMAT(I2,7X,3I2,3I1,9X,6A2,3X,I2)
  501 FORMAT(1H0)
  510 FORMAT(/,50X,'*** VOLUME SUMMARY ***')
  515 FORMAT(//,40X,'COURSE, ',6A2,5X,'VOLUME NO.  ',A2,/,20X,'NOTE,
     1THE ** SIGNIFIES THAT NO MATERIALS HAVE BEEN SUBMITTED FOR PROCESS
     1ING')
  520 FORMAT(1X,12A2,A1,3A2,A4,12(2X,A4,A2))
  521 FORMAT(A2,12A2,A1,2A2,A2,A3,12(A4,A2))
  530 FORMAT  (10X,'STUDENT',15X,'CSN  PRE.',5(3X,'STUDY'),
     1'  ASSIGN    HOME    POST   PERF.    CUM.   CAPL.',/,
     237X,'TEST   GDE(1)  GDE(2)  GDE(3)  GDE(4)  GDE(5)',
     311X,'WORK    TEST   INDEX    AVE.   INDEX')
  540 FORMAT(1X,12A2,A1,6X,I3,6X,'STUDENT DROPPED')
  550 FORMAT(1X,12A2,A1,6X,I3,6X,'THIS STUDENT DOES NOT HAVE ANY MATERIA
     1LS TO BE PROCESSED')
  999 RETURN
      END
/*
// LBLTYP NSD(5)
// EXEC LNKEDT
/&

00C10 IS IT EOF
REPLY Y
 END OF DATA

PAGE100
```